

Release 1.3

(Inklusive Release 1.3.16)
Addendum zum Bedienerhandbuch V1.0
6.3.2009

Hinweis

Die Angaben in dieser Dokumentation entsprechen dem Stand der Entwicklung zur Zeit der Drucklegung und sind daher unverbindlich.
NTI AG behält sich vor, Änderungen, die dem technischen Fortschritt bzw. Der Produkteverbesserung dienen, jederzeit und ohne Angaben von Gründen vorzunehmen. Im Übrigen verweisen wir auf unsere "Allgemeinen Geschäftsbedingungen" in der jeweils gültigen Ausgabe

Version 1.20 / 6. März 2009



1. Neuerungen in der *LinMot®* Software

1.1 Übersicht

Für den Release 1.2 und 1.3 wurden diverse Erweiterungen an der PC-Software *LinMot®* Talk sowie an der Firmware, die auf den *LinMot®* Servo Controllern läuft, vorgenommen. Dieses Handbuch beschreibt alle neuen Funktionen und erklärt Unterschiede zum Release 1.1. Bei allen Änderungen wurde darauf geachtet, dass die Kompatibilität mit den Releases 1.0 und 1.1 bewahrt werden konnte.

Die folgende Liste gibt einen Überblick über die neuen Funktionen. Alle Neuerungen werden in separaten Kapiteln detailliert beschrieben.

Neue Funktionen im Release 1.3.16

- Die Servokontroller Version 3 (E1001, E2001, E4001, E2031 und E4031) sind unterstützt. Diese Kontroller sind auch mit integrierter Masterencoderschnittstelle verfügbar (kein Zusatzmodul mehr nötig).
- Die High Performance Motoren P0x-23F-HP und P0x-37-HP sind unterstützt.
- Die untermassigen Läufer in den Motorkombinationen PS0x-37 mit PL01-19, PS0x-37-HP mit PL01-19, PS0x-37F mit PL01-19 und PS0x-48 mit PL01-27 sind unterstützt.

Neue Funktionen im Release 1.3.15

- Die Motorenserie P0x-48 wird mit verbesserten Positionstabellen unterstützt.
- Der Kurventyp “Limited Jerk” besitzt nun einen eigenen Parameter für den Ruck. Siehe Kapitel 1.5 “Rucklimitierte Bewegungsprofile”.

Neue Funktionen im Release 1.3.14

- Die Motorenserie P0x-48 wird unterstützt.
- Geräuschoptimierungsfilter für Motoren im Stillstand hinzugefügt. Siehe Kapitel “Regel- und Steuer-Parameter” auf Seite 81.
- CANopen-Interface hinzugefügt. CANopen Servo-Controller sind als eigenständige Produkttypen erhältlich, wie z.B. auch DeviceNet-Controller.
- Befehle GotoPositionFromActualPosition und SetDemandPositionToActualPosition hinzugefügt in den Interfaces MT, ASCII, DP, DN and CO (CANopen).
- ASCII: Der Befehl !EX (lese Zustands-Flags) beinhaltet neu auch die Unterzustands-Information NotStop.

Neue Funktionen im Release 1.3.12

- Beim Einloggen prüft *LinMot®*-Talk die Releasekonsistenz zwischen PC- und Kontroller-Software.
- Die RS485-Schnittstelle unterstützt bei beiden Protokollen (RS-Talk und ASCII RS485) den half wie auch den full duplex Modus.

Neue Funktionen im Release 1.3.11

- Der ASCII-Befehlssatz wurde erweitert mit Befehlen um Kurven und Cams von der aktuellen Position aus zu starten. Ebenfalls wurden Befehle für einen Variablen-Setup-Service hinzugefügt. Siehe Kapitel 8.3 “Commands overview”
- Es gibt neue MT-Befehle um Kurven und Cams von der aktuellen Position aus zu starten. Siehe Kapitel 2.3 “Die Zustandstabelle”.



- Beim Profibus DP gibt es neu Module für Fehler und Warnungen. Siehe Kapitel 3.7 “Datenmodul-Übersicht”
- Das Oszilloskop in der *LinMot®* Talk Software kann nun auch im Offline-Modus abgespeicherte Aufzeichnungen anzeigen.

Neue Funktionen im Release 1.3.10

- Mit dem neuen **Tuning Tool** können die **FeedForward-Parameter** (wie Current Offset, FF_Acceleration und FF_Deceleration) entsprechend dem Motorentyp, der Lastmasse, Reibung etc. sehr einfach eingestellt werden. Eine detaillierte Beschreibung des Tuning Tools ist im Kapitel 7.7 “Das Tuning Tool” zu finden.
- Aufzeichnungen mit dem *LinMot®*-Oszilloskop können gespeichert und wieder geladen werden. Dies macht einen Austausch sehr einfach. Siehe Kapitel 1.2 “Abspeichern von Oscilloskopaufzeichnungen”.
- Der **Error Inspector** zeigt zusätzlich den **Betriebszustand** des Kontrollers und die sog. **Logged Warnings** an. Dies bedeutet, dass alle Warnungen, welche seit der letzten Transition in den RUN Zustand aufgetreten sind, angezeigt werden. Diese Funktion kann für Inbetriebnahmen sehr hilfreich sein. Siehe Kapitel 1.3 “Logged Warnings”.
- Die Unterstützung des sog. **Gantry** Modus macht das Initialisieren und Fahren mit Anordnungen, wo zwei oder mehr Motoren mechanisch verbunden zusammen arbeiten, aber jeder Motor einzeln positionsgeregelt werden muss. Die Kombination mit externen Positionssensoren und Booster-Motoren ist möglich. Siehe Kapitel 5.2 “Master/Gantry operation”.
- Langhubige Motoren könenn nun über den vollen Hubbereich ohne die Home-Position zu verschieben betrieben werden. Hierfür gibt es in der Sensorkonfiguration den Modus **Internal Sensor 40µm**. In dieser Einstellung ist ein Positionsinkrement 40 µm, was einen Gesamthub von **2520mm** erlaubt. Siehe Kapitel “Positionssensorik” auf Seite 73.
- Bei gewissen Applikationen ist es hilfreich, wenn alle Motoren die Initialisierungsfahrt miteinander ausführen, weshalb die Systemeinstellung **Init Together** hinzugefügt wurde. Siehe Kapitel 6.2 “System-Parameter”.
- Neu werden **A/B Sensoren** für die Positionserfassung unterstützt, welche am **Masterencoder Modul** angeschlossen werden. Bis zu zwei A/B Sensoren können für beliebige Motoren angeschlossen werden. Siehe Kapitel 4.2 “A/B sensors”
- Release 1.3.10 unterstützt die **Masterencoder**-Funktionalität (siehe Addendum Master Encoder) zusammen mit den Schnittstellen MT (Multitrigger), DP (Profibus) oder DN (DeviceNet). Für jedes Interface wurden masterencoderspezifische Befehle für Rezeptur, Kurvenscheibenstart oder Umschalten auf Zeitkurven hinzugefügt.
- Das Konfigurations- und Debugprotokoll **RS-Talk** kann nun über **RS232** und **RS485** betrieben werden. Die beiden Schnittstellen werden komplett unabhängig voneinander bedient, was eine gleichzeitige Benutzung erlauft.

Neue Funktionen im Release 1.3.9

- Ab Release 1.3.9 wird der neue Kurventyp **Limited Jerk** im **Curve Creator** unterstützt. Bei diesem Typ werden wie bei **Point To Point** die maximale Geschwindigkeit und Beschleunigungen vorgegeben. Die Beschleunigung (und somit die Kraft) ändert sich aber nicht sprungartig, sondern mit einer Rampe.



- Der Kurventyp **Ramp** wird nicht mehr unterstützt. Er kann einfach mit dem Typ **Manual** mittels zwei Punkten und der Zeitangebe ersetzt werden. Mit dem **Curve Inspector** importierte Kurven werden automatisch auf den Typ **Manual** konvertiert.

Neue Funktionen im Release 1.3

- Parallel-Betrieb von bis zu vier Linearmotoren (Master/Booster). Damit kann auf einfache Art die Kraft, die für eine Bewegung zur Verfügung steht, erhöht werden.
- Unterstützung für externe Sensorik, welche eine wesentlich bessere Genauigkeit ermöglicht. So ist es möglich, mit dem entsprechenden Magnetband auf 10 µm genau zu positionieren.
- Die *LinMot®* Software unterstützt die neuen PROFIBUS Servo Controller: E130-DP, E230-DP, E430-DP, E1030-DP, E2030-DP und E4030-DP.
- Mit der PC-Software Release 1.3 können sowohl Servo Controller mit Firmware Release 1.2 als auch Servo Controller mit dem neuen Release 1.3 parametriert und bedient werden.
- Der Zustand **DISABLE** wird nun durch kurzes, zweimaliges Blinken der beiden LEDs **Stat A** und **Stat B** signalisiert. Er unterscheidet sich nun eindeutig vom Zustand **STOP**. Das komplette Zustandsdiagramm mit allen LED-Kodes ist in Abbildung 1-9, “Betriebszustände mit LED-Anzeige für Version 2 Kontroller”, auf Seite 11 dargestellt.
- Die Multitrigger Servo Controller unterstützen die vier neuen Befehle **Set PID**, **Set FF**, **Set Cur. Offset** und **Set CP**. Mit Hilfe dieser Befehle können die Einstellungen des Reglers und der Bewegungsprofile während dem Betrieb geändert werden.
- Neue ASCII-Befehle zum Starten von Bewegungen mit Hilfe eines Triggerpuls und zum Schreiben und Lesen von diversen Parametern. Eine ausführliche Beschreibung findet sich im Kapitel “LinMot®ASCII-Protokoll” auf Seite 117.
- Die Geschwindigkeit, Amplitude und der Positionsoffset von Bewegungsprofilen können im Betrieb über das ASCII-Protokoll oder den PROFIBUS eingestellt werden.

Neue Funktionen im Release 1.2

- Unterstützung eines neuen Kurventyps für die Erstellung von ruckbegrenzten Positionsprofilen. Mit Hilfe dieses Kurventyps können sehr einfach Positionsprofile erstellt werden, die eine lineare Zunahme der Beschleunigung aufweisen.
- Der sog. **Package Installer** erleichtert die Aufrüstung von Servo Controllern auf den neusten Release. Mit nur wenigen Mausklicks kann ein Servo Controller, der mit dem Release 1.0 oder 1.1 programmiert ist, auf den Release 1.2 aufgerüstet werden.
- Positionsregler: der interne Positionsregler wurde optimiert und mit neuen Funktionen ausgestattet. Durch diese Erweiterungen kann der Regler bei anspruchsvollen Servoanwendungen besser an die Applikation angepasst werden.



- Implementation eines ASCII-Protokolles für die RS232- und RS485-Schnittstelle. Mit Hilfe dieses Protokolles können sehr einfach Anwendungen programmiert werden, die mehrere *LinMot®* Motoren über eine RS232- oder RS485-Schnittstelle ansteuern. Über das Protokoll können sowohl vordefinierte Positionsprofile gestartet, wie auch beliebige Sollpositionen angefahren werden. Das neue ASCII-Protokoll ist im Kapitel “LinMot®ASCII-Protokoll” auf Seite 117 dieses Handbuchs beschrieben.
- Unterstützung des Betriebs von *LinMot®* P01-23x160 Motoren mit den Servo Controllern der Serie E1000.
- Importieren von Motorkonfigurationsdaten von unterschiedlichen Servo Controllern wird unterstützt. Beispielsweise kann nun die Motorkonfiguration eines E100 Servo Controllers problemlos auf einen E200, E400, E1000, E2000 oder E4000 Servo Controller geladen werden.
- Die Multitrigger Servo Controller unterstützen zwei neue Befehle. Mit dem Befehl **Redefine Position** kann die Istposition des betreffenden Motors neu definiert werden. Mit dem Befehl **Set Current** kann der Maximalstrom und somit die Kraft eines Motors im Betrieb limitert werden.
- Im **Error Inspector** ist neu eine Funktion zur Anzeige der anliegenden I/O-Werte verfügbar. Mit Hilfe dieser I/O-Statusanzeige können Probleme bei der Inbetriebnahme effizient gelöst werden.
- Unterstützung der Einstellung ‘Grosse Schriften’ in Windows 95/NT. Benutzer von grossen Monitoren können nun die Einstellung ‘Grosse Schriften’ benutzen. Diese Einstellung garantiert, dass alle Schriften auf dem Bildschirm, auch bei der Benutzung von 19” oder 21” Zoll Monitoren, lesbar bleiben.



1.2 Abspeichern von Oscilloskopaufzeichnungen

Das Oszilloskop unterstützt neu das Speichern und Öffnen von Aufzeichnungen. Mit der Taste “Save Configuration” wird eine komplette Oszilloskop-Konfiguration mit den aufgezeichneten Werten abgespeichert als “.ose”-Datei. Beim Öffnen einer solchen Konfiguration werden die Werte angezeigt.

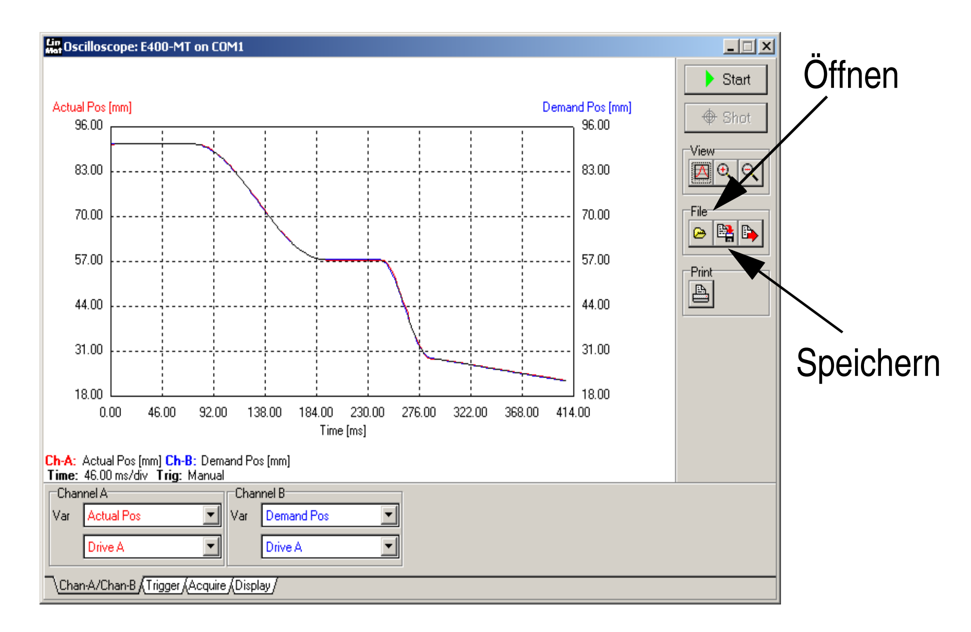


Abbildung 1-1: Speichern und Öffnen von Oszilloskop-Aufzeichnungen

1.3 Logged Warnings

Der Servocontroller speichert alle Warnungen, welche seit dem letzten Wechsel in den Zustand RUN aufgetreten sind. Diese sog. “logged warnings” können mit dem **Error Inspector** beim Klicken auf die Taste **Warnings** angezeigt werden.

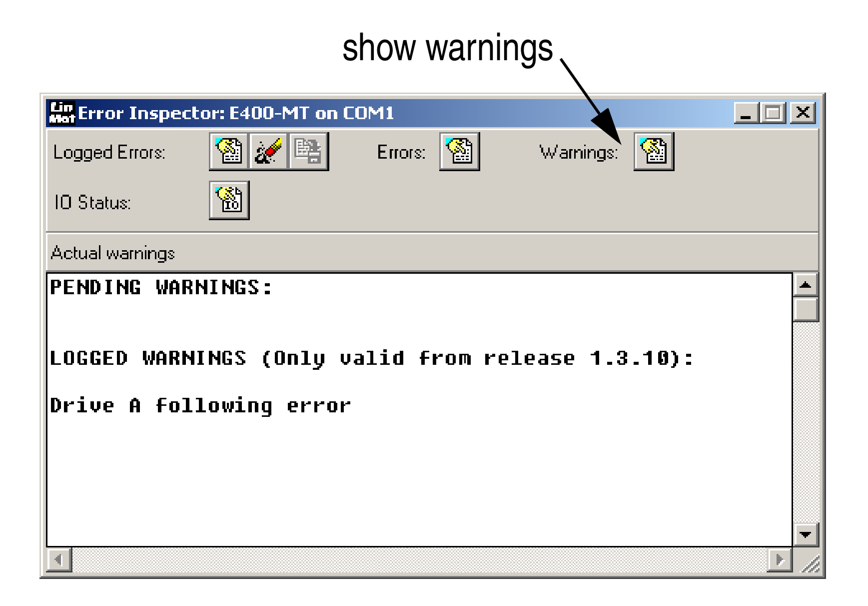


Abbildung 1-2: Anzeigen von Warnungen



1.4 Ruckminimierte Bewegungsprofile

Die Erstellung von ruckminimierten Bewegungsprofilen erfordert viel manuelle Rechenarbeit. Um die Erstellung von solchen Bewegungsprofilen zu vereinfachen, wurde der **Curve Editor** mit einem neuen Wizard, einem Hilfswerkzeug, ausgerüstet. Der Wizard berechnet, unter Berücksichtigung der Parameter Startpunkt, Endpunkt und der gewünschten Verfahrzeit, das Bewegungsprofil mit dem kleinsten möglichen Ruck. Die Abbildung 1-3, “Weg, Geschwindigkeit und Beschleunigung eines ruckminimierten Bewegungsprofiles” zeigt wie der Weg, die Geschwindigkeit und die Beschleunigung bei einem solchen Profil aussehen.

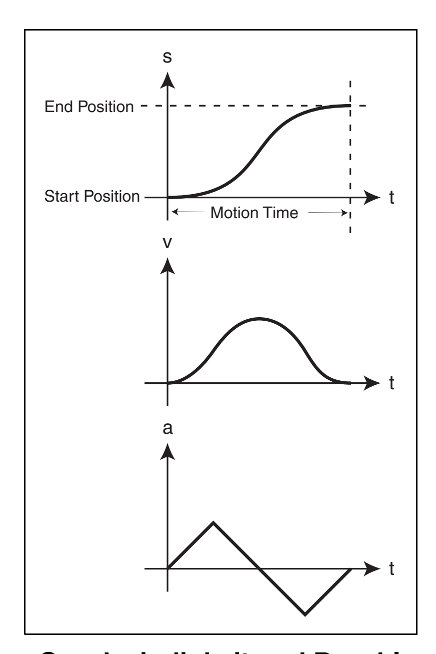


Abbildung 1-3: Weg, Geschwindigkeit und Beschleunigung eines ruckminimierten Bewegungsprofiles

Die Abbildung 1-2, “Beispiel eines ruckminimierten Bewegungsprofils”, auf Seite 3 zeigt den Kurveneditor, mit dem ein ruckminimiertes Bewegungsprofil erstellt wurde. Das ruckminimierte Bewegungsprofil ist im Curve Creator unter der Bezeichnung ‘Minimal Jerk’ zu finden.

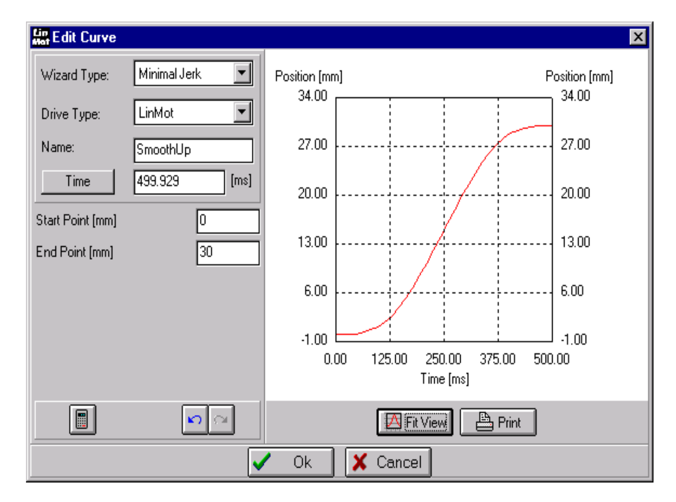


Abbildung 1-4: Beispiel eines rucklimitierten Bewegungsprofils



1.5 Rucklimitierte Bewegungsprofile

Das rucklimitierte Bewegungsprofil besitzt nun einen zusätzlichen Parameter für den Ruck. Die Releases 1.3.9 bis 1.3.14 haben automatisch eine dreiecksförmige Beschleunigung angenommen.

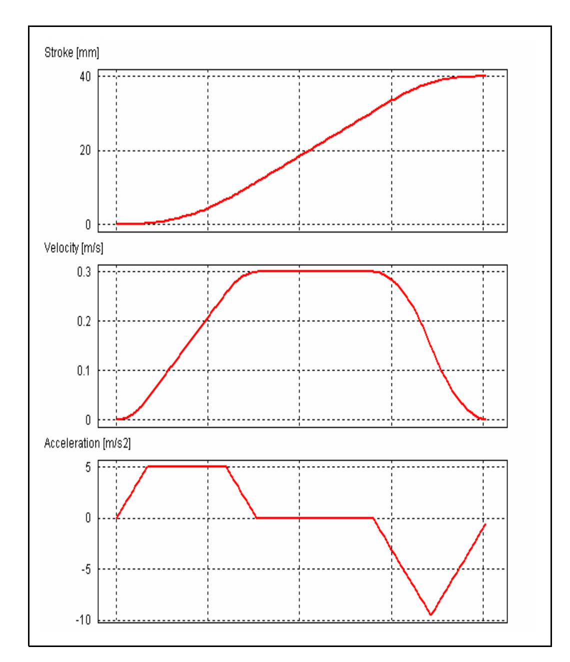


Abbildung 1-5: Weg, Geschwindigkeit und Beschleunigung eines rucklimitierten Bewegungsprofiles

Die Abbildung 1-6, “Beispiel eines rucklimitierten Bewegungsprofils”, auf Seite 7 zeigt den Kurveneditor, mit dem ein rucklimitiertes Bewegungsprofil erstellt wurde. Kurven vom Typ “Limited Jerk”, welche mit den Versionen 1.3.9 bis 1.3.14 erzeugt wurden, werden beim Importieren automatisch in “Manual”-Typen konvertiert.

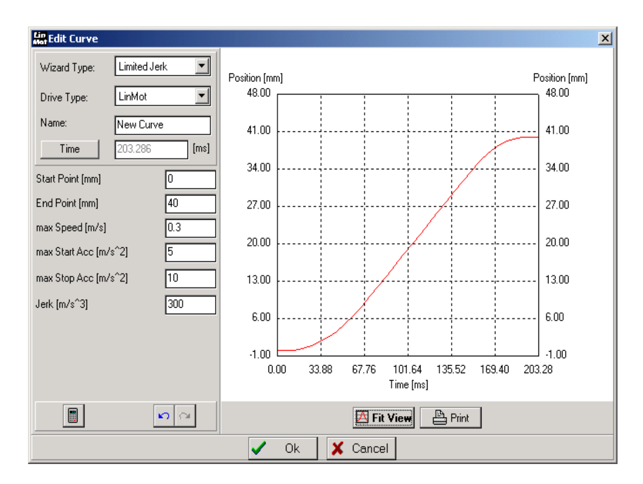


Abbildung 1-6: Beispiel eines rucklimitierten Bewegungsprofils

1.6 Package Installer

Das Aufrüsten von Servo Controllern auf einen neuen SW-Release war bis anhin mit dem Laden von vielen einzelnen Dateien auf den Controller verbunden. Mit dem neuen **Package Installer** kann ein Release nun mit wenigen Mausklicks auf den Controller geladen werden. Dabei stellt der **Package Installer** fest, um welchen Controllertyp es sich handelt und lädt die Software für den jeweiligen Typ automatisch.

Der **Package Installer** wird durch das Anwählen des Menüpunkts ‘Special’ -> ‘Install Package’ gestartet. Danach muss als User-ID **Install** und als Passwort **NTI**, in Grossbuchstaben, eingegeben werden. Die Installationsdatei befindet sich im Verzeichnis **Lin...\Firmware**.

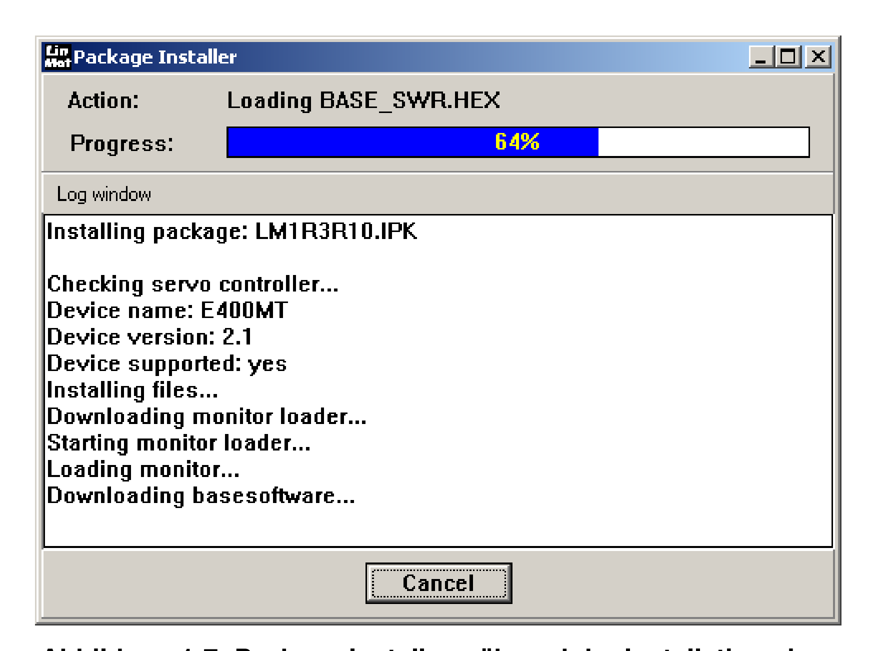
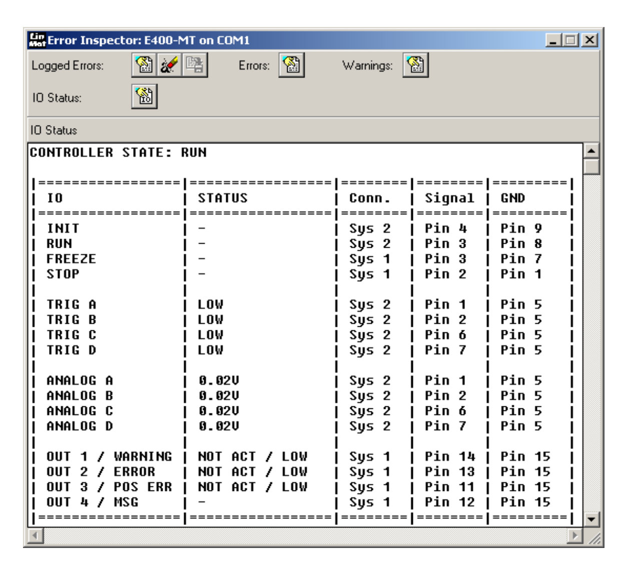


Abbildung 1-7: Package Installer während der Installation einer *LinMot*® E400-MT Servo Controller



1.7 I/O-Statusanzeige

Die neue I/O-Statusanzeige unterstützt den Anwender bei der Inbetriebnahme. Sie zeigt den Zustand der Eingangs- und Ausgangssignale des angeschlossenen Servo Controllers an. Die Statusanzeige wird durch das Drücken auf die Taste „IO Status“ im **Error Inspector** aktiviert.



IO	STATUS	Conn.	Signal	GND
INIT	-	Sys 2	Pin 4	Pin 9
RUN	-	Sys 2	Pin 3	Pin 8
FREEZE	-	Sys 1	Pin 3	Pin 7
STOP	-	Sys 1	Pin 2	Pin 1
TRIG A	LOW	Sys 2	Pin 1	Pin 5
TRIG B	LOW	Sys 2	Pin 2	Pin 5
TRIG C	LOW	Sys 2	Pin 6	Pin 5
TRIG D	LOW	Sys 2	Pin 7	Pin 5
ANALOG A	0.02V	Sys 2	Pin 1	Pin 5
ANALOG B	0.02V	Sys 2	Pin 2	Pin 5
ANALOG C	0.02V	Sys 2	Pin 6	Pin 5
ANALOG D	0.02V	Sys 2	Pin 7	Pin 5
OUT 1 / WARNING	NOT ACT / LOW	Sys 1	Pin 14	Pin 15
OUT 2 / ERROR	NOT ACT / LOW	Sys 1	Pin 13	Pin 15
OUT 3 / POS ERR	NOT ACT / LOW	Sys 1	Pin 11	Pin 15
OUT 4 / MSG	-	Sys 1	Pin 12	Pin 15

Abbildung 1-8: IO-Statusanzeige

1.8 Neue Befehle für MT Servo Controller

Redefine Position

Definiert Istposition

Mit diesem Befehl wird die Istposition neu definiert. Dieser Befehl darf nur ausgeführt werden, wenn der betreffende Aktor die Sollposition erreicht hat und sich nicht im Zustand **FREEZE** befindet.

Set Current

Setzt Maximalstrom

Dieser Befehl setzt den Maximalstrom und somit die maximale Kraft des betreffenden Aktors neu. Es dürfen nur positive Werte vorgegeben werden.

Set FF

Setzt Feed Forward-Parameter (Release 1.3)

Dieser Befehl setzt die **Feed Forward**-Parameter im Positionsregler. Der Befehl wird verwendet, um bei einer Veränderung der Lastmasse eine optimale Anpassung des Reglers zu erreichen.

Set PID

Setzt PID-Parameter (Release 1.3)

Dieser Befehl setzt die PID-Parameter im Positionsreglers. Der Befehl kann gebraucht werden, um den Regler während des Betriebs optimal anzupassen.

Set Cur. Offset

Setzt Stromoffset (Release 1.3)

Dieser Befehl setzt den Stromoffset für Linearmotoren.

Set CP

Setzt Kurven Parameter (Release 1.3)

Dieser Befehl setzt die Kurven-Parameter Offset, Amplitude und Geschwindigkeit.

Recipe

Wechsle Rezeptur für Master Encoder (Release 1.3.10)

Dieser Befehl dient zur Rezepturwahl beim Masterencoder-Betrieb für den nächsten Kurvenscheiben-Zyklus. Dieser Befehl ist nur im Masterencoder-Modus verfügbar.

Start Cam

Wechsle auf Kurvenscheibenmodus (Release 1.3.10)

Dieser Befehl schaltet einen Motor auf den Kurvenscheibenmodus um. Im Betrieb kann zwischen Kurvenscheiben- und zeitbasiertem Modus beliebig umgeschaltet werden.Dieser Befehl ist nur im Masterencoder-Modus verfügbar.

Die folgende Tabelle gibt einen Überblick über die verfügbaren MT-Befehle:

MT-Befehle	
Befehl	**Beschreibung**
No Operation	Keinen Befehl ausführen
Abs. Position	Positionierung auf absolute Position
Rel. Position	Verschiebung um Positiondifferenz
Abs. Current	Setzt absoluten Stromwert
Rel. Current	Setzt relativen Stromwert
Set Current	Maximalstrom setzen
Set FF	Setzt FF-Reglerparameter
Set PID	Setzt PID-Reglerparameter
Set Cur. Offset	Setzt Stromoffset
Set CP	Setzt Kurvenparameter
Curve	Sollwertkurve abarbeiten
Move Home Position	Verschieben der Referenzposition
Redefine Position	Definieren der Istposition
Freeze / Unfreeze	Bewegung unterbrechen
Stop	Bewegung stoppen
Recipe	Wechsle Rezeptur für Master Encoder
Start Cam	Wechsle auf Kurvenscheibenmodus

Abs Pos Act Pos

Positionierung auf absolute Position von aktueller Position (ab Release 1.3.13)

Fast gleicher Berfehl wie Abs.Position, nur dass das Geschwindigkeits/Beschleunigungs-Filter von der aktuellen Position startet. Dieser Befehl kann beim Wegfahren von einer Pressposition benutzt werden.

SetDPostoAPos

Setze Sollposition auf Istposition (ab Release 1.3.13)

Dieser Befehl setzt die Sollposition auf die Istposition. Dieser Befehl wir z.B. gebraucht, wenn beim löschen des Freeze der Motor die Bewegung nicht mehr zu Ende fahren soll, oder wenn der Motor stromlos war und nur bei Bestromung keine Bewegung machen darf.



1.9 Betriebszustände

Der Zustand **DISABLE** wird nun durch kurzes, zweimaliges Blinken der beiden LEDs **Stat A** und **Stat B** signalisiert. Er unterscheidet sich nun eindeutig vom Zustand **FIRMWARE STOPPED**. Das folgende Zustandsdiagramm zeigt alle Zustände und die Bedingungen, welche für einen Zustandswechsel erfüllt sein müssen. Zusätzlich ist ab Release 1.3.9 der Zustand **STOP** durch Blink Codes von **INIT** und **RUN** unterscheidbar. Mit dem Zustand STOP ist Notstop gemeint. Dieser Zustand kann nur durch Rücksetzen der INIT, RUN, und STOP Requests nach **DISABLE** verlassen werden.

FIRMWARE STOPPED
Power Down | Stop Firmware
Power Up | Start Firmware
SETUP
WAIT FOR DISABLE
DRIVE INIT
!FR
FREEZE
FR
EP
!RR & !IR & !SR
IR
!IR
1.)
SR
ERROR
EP
DISABLE
!RR & !IR & !SR
STOP
!EP & !RR & !IR & !SR
1.)
EP
RR & !DIND & !WP
EP
!RR
SR
RR & !DIND &!WP & !IR
RUN
!RR & IR
EP
!FR
FREEZE
FR

Input signals have to be activated in the parameter tree!

RUN signal is set in state SETUP. For changing into the state DISABLE, the RUN signal must be cleared from the active interface!

LEDs: Fault, Ready, Stat A, Stat B
LED off
LED blinking
LED on

1.) From Release 1.3.9

Input Signals	Abbreviation
Set INIT Request	IR
Set RUN Request	RR
Set STOP Request	SR
Set FREEZE Request	FR
Reset INIT Request	!IR
Reset RUN Request	!RR
Reset STOP Request	!SR
Reset FREEZE Request	!FR

Internal Signals	Abbreviation
DRIVE_INIT_NOT_DONE	DIND
ERROR_PENDING	EP
WARNING_PENDING	WP

Abbildung 1-9: Betriebszustände mit LED-Anzeige für Version 2 Kontroller

FIRMWARE STOPPED (-)

Power Down | Stop Firmware

Power Up | Start Firmware

SETUP

WAIT FOR DISABLE (0)

DRIVE INIT (9)

!FR FREEZE FR

EP

!RR & !IR & !SR

IR

!IR

SR

ERROR (Exxx)

EP

DISABLE (1) ((2)if INIT DONE

!RR & !IR & !SR

STOP (7)

!EP & !RR & !IR & !SR

EP

RR & !DIND & !WP

EP

!RR

SR

RR & !DIND &!WP & !IR

RUN (8)

!RR & IR

EP

!FR FREEZE FR

Input Signals	Abbreviation
Set INIT Request	IR
Set RUN Request	RR
Set STOP Request	SR
Set FREEZE Request	FR
Reset INIT Request	!IR
Reset RUN Request	!RR
Reset STOP Request	!SR
Reset FREEZE Request	!FR

Internal Signals	Abbreviation
DRIVE_INIT_NOT_DONE	DIND
ERROR_PENDING	EP
WARNING_PENDING	WP

Input signals have to be activated in the parameter tree!

RUN signal is set in state SETUP. For changing into the state DISABLE, the RUN signal must be cleared from the active interface!

Abbildung 1-10: Betriebszustände mit 7-Segment-Anzeige für Version 3 Kontroller



Aufstartvorgang

Das folgende Diagramm zeigt die Ein- und Ausgangssignale bei einem typischen Aufstartvorgang.

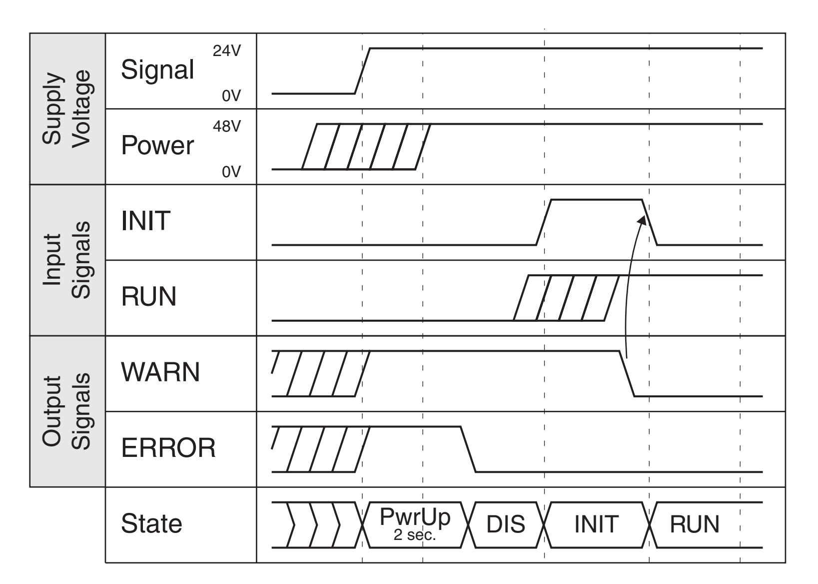


Abbildung 1-11: Aufstartvorgang

Unterbruch der Leistungsversorgung

In vielen Fällen ist es wegen den Sicherheitsbestimmungen erforderlich, dass bei einem Notaus die Leistungsversorgung der Motoren unterbrochen wird. Die *LinMot®* Steuerungen ermöglichen dies durch eine getrennte Speisung. So kann bei einem Notaus der Signalteil der Controller weiterhin gespiesen werden aber die Leistungsversorgung kann unterbrochen werden, ohne dass die Motoren erneut initialisiert werden müssen. Das folgende Schema zeigt die relevanten Ein- und Ausgangssignale bei einem solchen Vorgang.

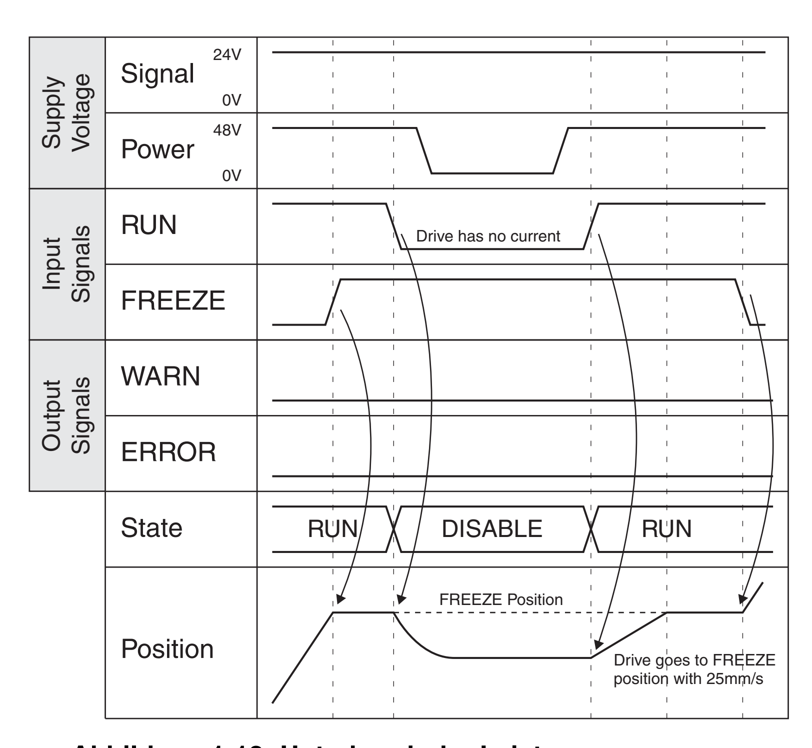


Abbildung 1-12: Unterbruch der Leistungsversorgung

2. MT Servo Controller

2.1 Systemübersicht

Die Synchronisation elektrischer Antriebe erfolgt allgemein über aufwendige elektronische Winkelübertragungen. Einfache SPS-Systeme sind hingegen rein ablaufgesteuert und bieten diese Funktionalität nicht. Neue Antriebskonzepte ermöglichen dennoch die Synchronisation von komplexen Bewegungen.

Mechatronische Steuerungskonzepte

Elektronische Hauptwelle

Bei der elektronischen Hauptwelle werden ausgehend von einem Winkelsignal, welches in den meisten Fällen von einem Winkelencoder der Hauptwelle stammt, alle Antriebe winkelsynchron, d.h. positionsgeregelt nachgeführt. Dabei kommen sowohl zentrale als auch dezentrale Konzepte zum Einsatz (Bild 1). Im letzteren Fall wird jeder dezentrale Antrieb mit einem Winkelsignal versorgt, der es ihm ermöglicht, die geforderte Sollposition aus einer abgespeicherten Tabelle auszulesen. Sieht man einmal von Schleppfehlern ab, kann die gesamte Maschine als starr miteinander verkoppelt betrachtet werden. Der steuerungstechnische Aufwand derartig realisierter Maschinen ist nicht zu unterschätzen und bedarf aufwendiger Elektronik. Insbesondere muss unterschieden werden zwischen dem im Betrieb geforderten synchronem Lauf der Antriebe, welcher lediglich durch Schleppfehler gestört wird, und den als 'spezielle Bedingungen' bekannten Situation, wie sie beim Einschalten der Maschine, im Wartungsfall oder bei Störungen auftreten.

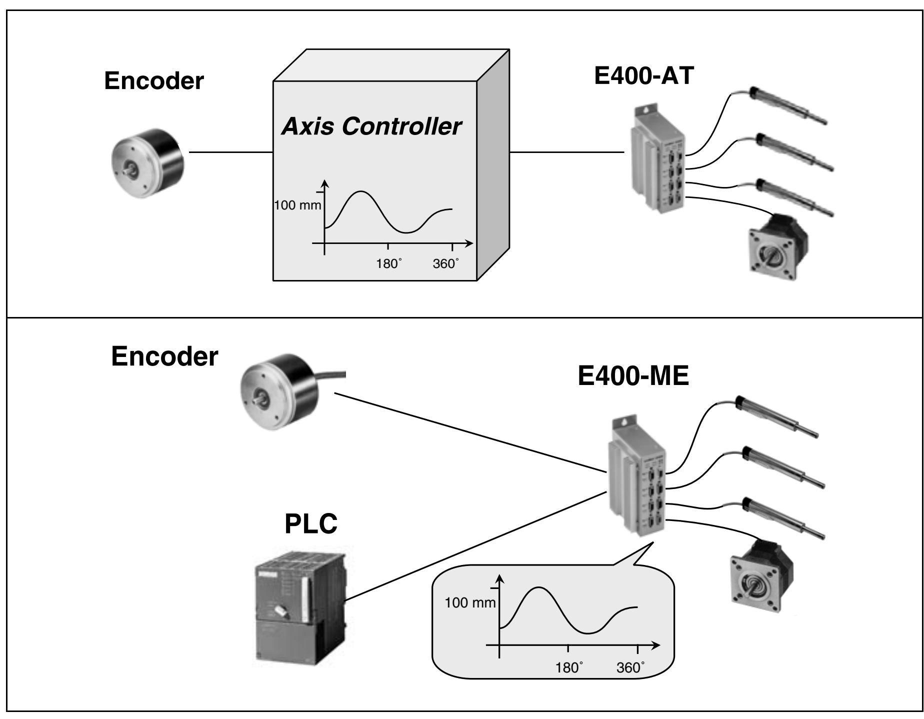


Abbildung 2-1: Elektronische Synchronisation

Bei der elektronische Hauptwelle werden alle Antriebe auf Basis eines Winkelsignals eines Encoders aufeinander synchronisiert, wobei die Bewegungskurven zentral in einer Achssteuerung (Bild oben) oder dezentral (Bild unten) in den Antrieben selber abgespeichert sind.



Sequentielle Ablaufssteuerung

Eine grundsätzlich andere Philosophie verfolgt die mit einem Nockenschaltwerk vergleichbare Ablaufsteuerung: Hier wird die Steuerung der einzelnen Funktionen nicht auf der Basis einer Winkelinformation ausgeführt, sondern aufgrund sich folgender Ereignisse bzw. zeitlicher Abstände. Gerade in der Welt der SPS-Programmierung ist diese Betrachtungsweise weitverbreitet.

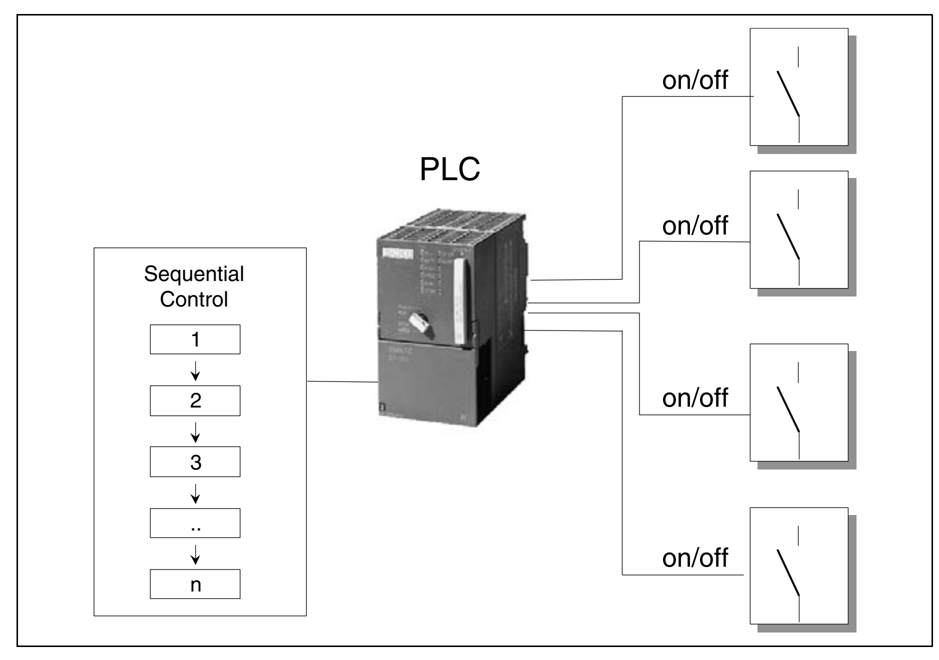


Abbildung 2-2: Ablaufsteuerung

Ablaufsteuerungen gehen Schritt um Schritt durch das Programm, wobei Ereignisse oder zeitliche Abstände die Übergänge zwischen den einzelnen Programmpunkten auslösen. Jeder Programmpunkt ist dabei ein in sich abgeschlossener Vorgang, wie z.B. das Ein- oder Ausschalten eines Relais.

Die Ablaufsteuerung weist gegenüber der winkelsynchronen Steuerung mittels elektronischer Welle eine wesentliche Vereinfachung auf, indem alle Betriebssituationen konzeptionell identisch behandelt werden. Konkret muss nicht unterschieden werden zwischen dem winkelsynchronem Lauf im Normalbetrieb und den sequentiellen Abläufen während der Initialisierung, der Wartung oder in Notfallsituationen. Einschränkungen treten hingegen überall dort auf, wo Vorgänge in speziellen Bewegungen parallel bzw. synchron zueinander ablaufen müssen. Ein typisches Beispiel sind mechanische Funktionseinheiten mit mehreren ineinandergreifenden Bewegungen zum Falten oder Montieren von Verpackungen und Teilen.

SPS kombiniert mit komplexen Bewegungsabläufen

Mit einfachen SPS-Systemen werden heutzutage eine Vielzahl von Steuerungsaufgaben im Maschinenbau gelöst. Sobald jedoch aufwendige bzw. kombinierte Bewegungen kontrolliert werden müssen, werden die Anforderungen an die SPS-Systeme derart hoch, dass keinesfalls mehr die preislich interessanten Kleingeräte eingesetzt werden können. Der mechatronische Maschinenbau führt nun dazu, dass die Zahl der durch Linearmotoren oder rotative Antriebe realisierten Funktionsmodule stetig steigt und somit immer mehr komplexe Bewegungsabläufe zu steuern sind. Aus dieser Erkenntnis heraus wurde das Antriebssystem *LinMot*® mit einer sogenannten Multitrigger-Funktionalität erweitert. Der Grundgedanke von 'Multitrigger' besteht darin, dass einzelne oder auch verkoppelte Bewegungen mehrerer Antriebe als in sich abgeschlossene Bewegungssequenzen verstanden werden. Das heisst, eine Bewegung bzw. ein Bewegungsablauf von mehreren verkoppelten Linear- oder Rotationsmotoren kann durch die SPS gestartet und gestoppt werden, vergleichbar einem Relais, das ein- und ausgeschaltet wird.

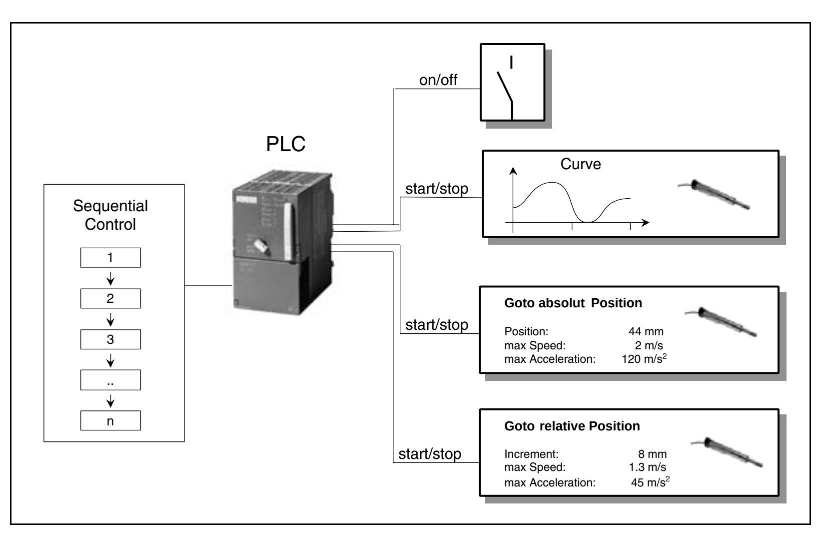


Abbildung 2-3: Ablaufsteuerung synchroner Bewegungen

Die Ablaufsteuerung geht Schritt um Schritt durch das Programm und löst vergleichbar dem Schliessen eines Relais komplexe Bewegungssequenzen (Absolutbewegungen, Relativbewegungen, Bahnkurven) aus, welche dann dezentral im Servo Controller ablaufen.

Als Bestätigung, dass die Bewegung ausgeführt wurde, hat die SPS die Möglichkeit, eine Rückmeldung im Sinne eines Endpositionsschalters oder eines Schleppfehlers zu überwachen. Zusätzlich besteht die Möglichkeit, dass die SPS eine gestartete Bewegung stoppt oder für eine bestimmte Zeit unterbricht. Ansonsten wird die gesamte Bewegungssequenz dezentral im Servo Controller ausgeführt und die SPS wird so komplett von den rechenintensiven Positionsregelungen entlastet. In oben stehender Abbildung sind verschiedene von *LinMot*® Multitrigger zur Verfügung gestellte Bewegungssequenzen mit den dazugehörenden Parametriermöglichkeiten dargestellt. Besonders interessant im Umfeld von kombinierten Bewegungen ist die Möglichkeit, pro *LinMot*® E400-MT Servo Controller bis zu 4 Antriebe gleichzeitig anzusteuern. Dadurch kann mit einem einzigen Start-Befehl der SPS die synchrone Bewegung von 4 Antrieben ausgelöst werden. Das nachfolgende Beispiel gibt einen Einblick in diese Möglichkeiten.



Einführen eines Deckels

Im folgenden wird schematisch ein Bewegungsvorgang dargestellt, wie er zum Einführen eines Deckels in einen Behälter notwendig ist.

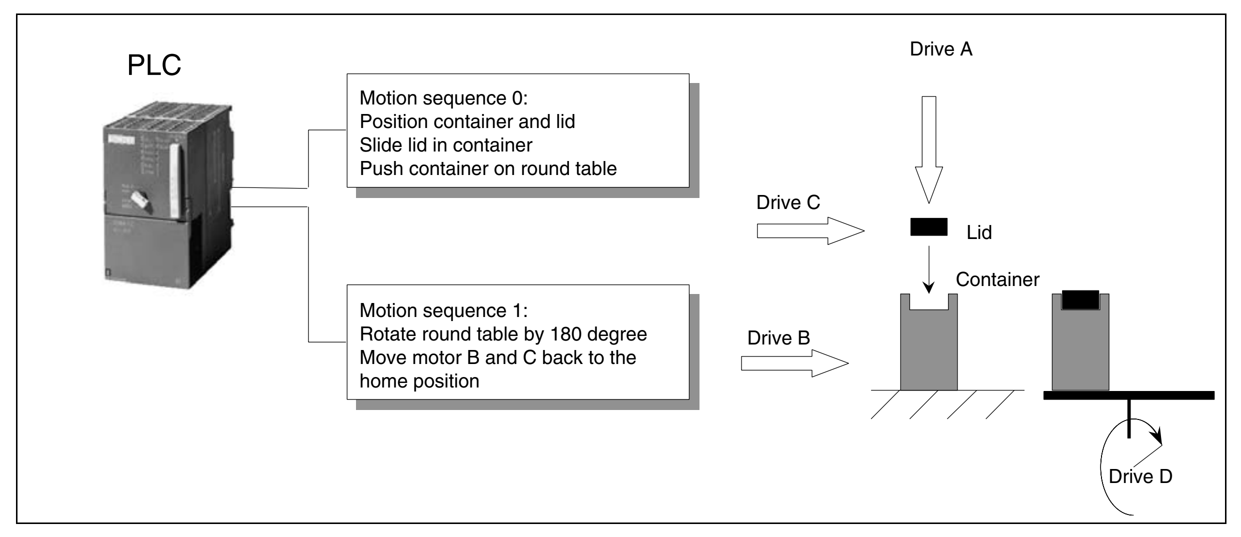


Abbildung 2-4: Einführen eines Deckels

Die beiden Bewegungssequenzen werden durch digitale Signale der SPS ausgelöst. Somit kontrolliert die SPS den gesamten Ablauf, ohne dass sie sich um die Synchronisation oder die eigentliche Positionsregelung der Antriebe kümmern muss.

Auf einen ersten Startbefehl der SPS bringen die Linearantriebe B und C den Behälter bzw. den Deckel in die Arbeitsposition, worauf gleichzeitig der Linearantrieb A mit einer vertikalen Bewegungskurve beginnt, welche den Deckel in den Behälter hinein schiebt. Anschliessend wird der Behälter auf einen Drehtisch hinaus gestossen und an die SPS ein entsprechendes Feedbacksignal ausgegeben (Bewegungssequenz 0). Wenn weitere Bedingungen erfüllt sind, startet die SPS die Bewegungssequenz 1, welche die Antriebe B und C in die Ausgangsposition zurück bewegt und gleichzeitig der Drehtisch (Antrieb D) um 180° dreht.



Die nächste Abbildung zeigt die Programmierung der Bewegungssequenzen 0 und 1 ('State 0' und 'State1') mit der *LinMot*® Multitrigger Steuerung sowie die Bewegungskurven (Curve 1, 2, 3) für die Bewegungssequenz 0.

State	Drive A - LinMot	Drive B - LinMot	Drive C - LinMot	Drive D - Stepper
0	**Curve** Curve number 1	**Curve** Curve number 2	**Curve** Curve number 3	**No Operation**
1	**No Operation**	**Abs. Position** Position 0 mm Speed 1 m/s Acc. 50.068 m/s^2	**Abs. Position** Position 0 mm Speed 1 m/s Acc. 50.068 m/s^2	**Rel. Position** Increment 180 Steps Speed 800.018 Steps/s Acc. 10014.306 Steps/s^

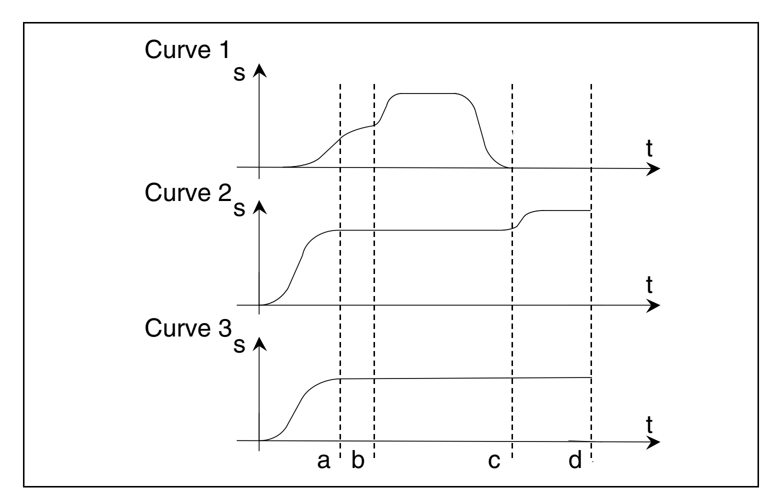


Abbildung 2-5: Programmierung der Bewegungssequenzen mit einem *LinMot*® MT Servo Controller

Auf der rechten Seite sind die Bewegungskurven der Bewegungssequenz 0 dargestellt: Diese sind wie folgt zu verstehen: Antrieb B (Curve 2) und C (Curve 3) bewegen den Behälter bzw. den Deckel zur Arbeitsposition (a). Gleichzeitig bewegt sich Antrieb A (Curve 1) vertikal zum Deckel hin (b), presst diesen in den Behälter und zieht sich anschliessend zur Ausgangsposition zurück (b bis c). Antrieb B (Curve 2) stösst anschliessend den Behälter auf den Rundtisch (c bis d).

Zusammenfassung

Das Multitrigger-Steuerkonzept ermöglicht zusammen mit Direktlinearantrieben und Schrittmotoren eine einfache Realisierung von komplexen mechatronischen Funktionseinheiten. Wesentlich dabei ist, dass die übergeordnete Gesamtsteuerung (SPS, PC) der Maschine von antriebstechnischen Aufgaben weitgehend entlastet ist und damit entsprechend einfach und kostengünstig ausgelegt werden kann.



2.2 Projektierung und Installation

In diesem Kapitel wird die Ansteuerung der *LinMot*® Multi Trigger Servo Controller von einer übergeordneten Steuerung eingehend erläutert. Die Multi Trigger Servo Controller der Serie Ex00-MT und Ex000-MT können im Multi Trigger Modus oder im Analog Trigger Modus betrieben werden.

Betrieb im Multi Trigger Modus

Der Multi Trigger Modus ist eine Erweiterung des Digital Trigger Modus auf dem AT Servo Controller (vgl. Benutzerhandbuch Kapitel 4.1.3). Im Digital Trigger Modus können mittels der steigenden und der fallenden Flanke des Triggersignals pro Motor zwei Sollwertkurven abgefahren werden.

Im Multi Trigger Modus können pro Motor bis zu 64 Sollwertkurven oder Sollpositionen auf dem Servo Controller gespeichert werden. Diese können mittels digitaler Triggersignale von der übergeordneten Steuerung aufgerufen werden.

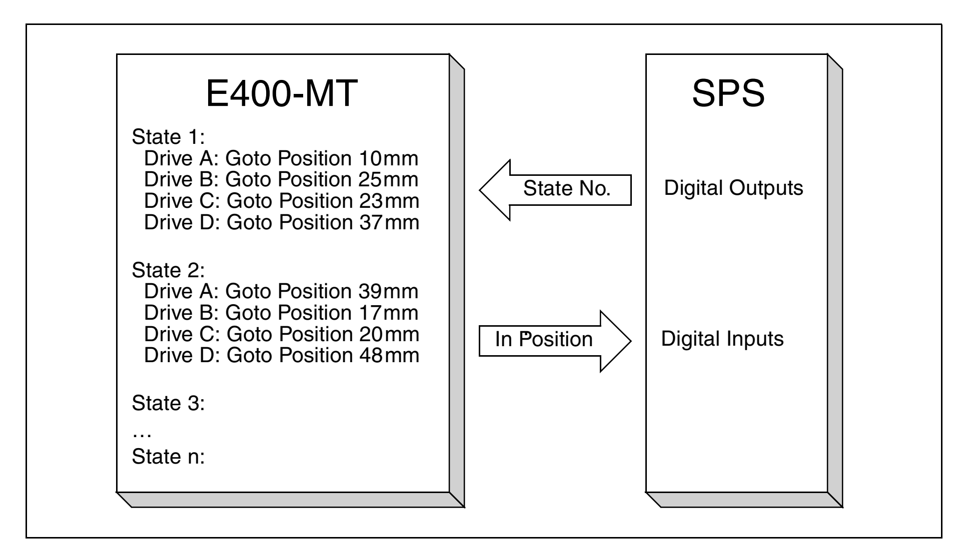


Abbildung 2-6: MT-Ansteuerung mittels digitaler Triggersignale

Die Bewegungen werden in Form von Zuständen (State 0, State 1….State 63) auf dem Servo Controller gespeichert. Die übergeordnete Steuerung (SPS) ruft die einzelnen Zustände mittels digitaler Triggersignal auf. Sobald die Aktoren die Endposition des aktuellen Zustands erreicht haben, wird dies der übergeordneten Steuerung mittels digitaler **In Position**-Signale mitgeteilt.

2.3 Die Zustandstabelle

In der Zustandstabelle können bis zu 64 Zustände (States) definiert werden. In jedem einzelnen Zustand werden die Bewegungen definiert, die der betreffende Aktor beim Aufruf des Zustands ausführen soll.

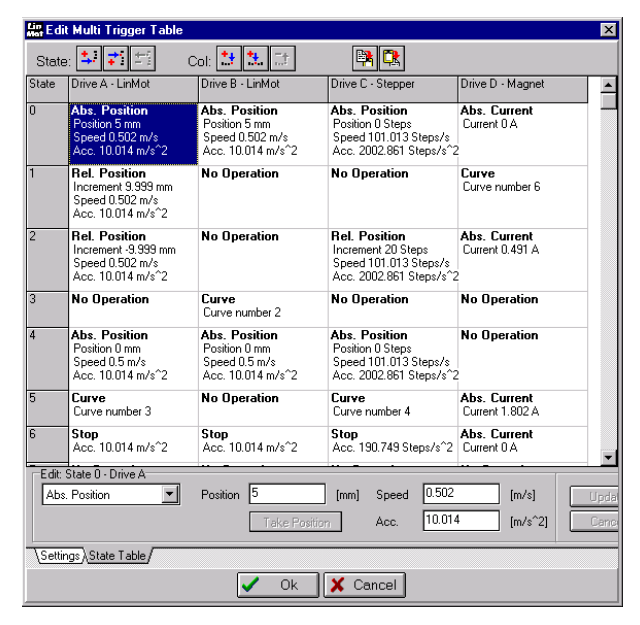


Abbildung 2-7: Die Zustandstabelle

Der auszuführende Befehl kann in der Zustandstabelle durch folgende Funktionen definiert werden:

No Operation

Aktor führt keine Bewegung aus

Der Aktor führt keine Bewegung aus oder führt eine bereits angefangene Bewegung zu Ende. Wird bei Servo Controllern für mehrere Achsen verwendet, wenn ein Aktor im betreffenden Zustand keine Bewegung ausführen oder die angefangene Bewegung zu Ende führen soll.

Abs. Position

Positionierung auf absolute Position

Der Aktor wird auf die gewünschte absolute Position (relativ zur Nullposition) unter Berücksichtigung einer einstellbaren max. Geschwindigkeit und max. Beschleunigung positioniert. Sobald der Aktor die Endposition erreicht hat, wird das "In Position" Signal aktiviert.

Abs. Current

Setzen eines absoluten Stromwertes

Dieser Befehl ist nur bei der Ansteuerung vom Magneten sichtbar und dient dazu, den Strom zu setzten.

Rel. Position

Verschiebung um relative Position

Der Aktor wird um die gewünschte Distanz (relativ zur aktuellen Sollposition) unter Berücksichtigung einer einstellbaren max. Geschwindigkeit und max. Beschleunigung verschoben. Sobald der Aktor die neue Sollposition erreicht hat, wird das "In Position" Signal aktiviert.

Rel. Current

Setzen eines relativen Stromwertes

Dieser Befehl ist nur bei der Ansteuerung vom Magneten sichtbar und dient dazu einen relativen Stromwert zu setzen.



Curve — **Sollwertkurve abarbeiten**

Ein gespeichertes Bewegungsprofil, das in Form einer Kurvendatei auf dem Servo Controller gespeichert, ist wird abgefahren. Sobald der Aktor den letzten Sollwert der Kurve erreicht hat, wird das Signal **In Position** aktiviert.

Move Home Position — **Verschieben der Referenzposition (Nullpunkt)**

Die Kurven und absolute Positionen beziehen sich immer auf die nach der Initialisierung ermittelte Referenzposition. Mit **Move Home Position** wird die Referenzposition (Nullpunkt) des betreffenden Antriebs um die gewünschte Distanz verschoben. Dieser Befehl darf nur ausgeführt werden, wenn alle Aktoren stillstehen und sich keiner der Motoren im Zustand **FREEZE** befindet.

Redefine Position — **Definieren der Istposition**

Mit diesem Befehl wird die Istposition neu definiert. Dieser Befehl darf nur ausgeführt werden, wenn der betreffende Aktor die Sollposition erreicht hat und sich nicht im Zustand **FREEZE** befindet.

Stop — **Bewegung stoppen**

Die momentane Bewegung wird abgebrochen und der Aktor wird unter Berücksichtigung einer einstellbaren max. Beschleunigung zum Stillstand gebracht. Sobald der Aktor zum Stillstand gekommen ist, wird das Signal **In Position** aktiviert.

Freeze/Unfreeze — **Bewegung unterbrechen**

Beim Freeze-Befehl wird die momentane Bewegung unterbrochen und der Aktor wird unter Berücksichtigung der für die aktuelle Bewegung gültigen max. Beschleunigung zum Stillstand gebracht. Im Unterschied zum Stop-Befehl wird das Signal **In Position** beim Freeze-Befehl nicht aktiviert.

Durch den Unfreeze-Befehl kann die unterbrochene Bewegung zu Ende geführt werden. Sobald die Bewegung zu Ende Geführt ist, wird das Signal **In Position** aktiviert. Werden Befehle aufgerufen, während dem sich der Aktor im Zustand **FREEZE** befindet, führt er beim Aufruf von **UNFREEZE** den zuletzt aufgerufenen Befehl aus (handelt es sich bei den zuletzt aufgerufenen Befehlen um **Rel. Position**-Befehle, werden die Relativpositionen addiert).

Beim Verlassen des Betriebszustandes **RUN** werden alle **Freeze**-Befehle aufgehoben.

Set Current — **Maximalstrom setzen**

Dieser Befehl setzt den Maximalstrom und somit die maximale Kraft des betreffenden Aktors neu. Es dürfen nur positive Werte vorgegeben werden.

Set Cur. Offset — **Stromoffset setzen**

Dieser Befehl setzt den Stromoffset. Der Stromoffset kann eingesetzt werden um eine statische Kraft, welche auf den Motor wirkt, zu kompensieren.

Set FF — **Setzen der Feed Forward-Parameter**

Dieser Befehl setzt die **Feed Forward**-Parameter. Er kann gebraucht werden um eine optimale Anpassung des Reglers bei einer Veränderung der Lastmasse zu erreichen.

Set PID — **Setzen der PID-Parameter**

Dieser Befehl setzt die PID-Parameter. Er kann gebraucht werden um den Regler während des Betriebs optimal anzupassen.

Set CP — **Setzen der Kurven-Parameter**

Dieser Befehl setzt die Kurven Geschwindigkeit, Amplitude und Offset.

Recipe — **Wechsle Rezeptur für Master Encoder (ab Release 1.3.10)**

Dieser Befehl dient zur Rezepturwahl beim Masterencoder-Betrieb für den nächsten Kurvenscheiben-Zyklus. Dieser Befehl ist nur im Masterencoder-Modus verfügbar.

Start Cam — **Wechsle auf Kurvenscheibenmodus (ab Release 1.3.10)**

Dieser Befehl schaltet einen Motor auf den Kurvenscheibenmodus um. Es kann zwischen Kurvenscheiben- und zeitbasiertem Modus beliebig umgeschaltet werden. Dieser Befehl ist nur im Masterencoder-Modus verfügbar.

Start Cam ActPos — **Wechsle auf Kurvenscheibenmodus und starte von aktueller Position (ab Release 1.3.11)**

Dieser Befehl schaltet einen Motor auf den Kurvenscheibenmodus um. Es kann zwischen Kurvenscheiben- und zeitbasiertem Modus beliebig umgeschaltet werden. Im Unterschied zum Start Cam Befehl wird der Parameter Curve Position Offset so gesetzt, dass die Kurve bei der aktullen Sollposition beginnt. Dieser Befehl ist nur im Masterencoder-Modus verfügbar.

Curve ActPos — **Wechsle auf Kurvenscheibenmodus (ab Release 1.3.11)**

Gleicher Befehl wie Curve aber der Parameter Curve Position Offset wird so gesetzt, dass die Kurve bei der aktuellen Sollposition beginnt.

Abs Pos Act Pos — **Positionierung auf absolute Position von aktueller Position (ab Release 1.3.14)**

Fast gleicher Berfehl wie Abs.Position, nur dass das Geschwindigkeits/Beschleunigungs-Filter von der aktuellen Position startet. Dieser Befehl kann beim Wegfahren von einer Pressposition benutzt werden.

SetDPostoAPos — **Setze Sollposition auf Istposition (ab Release 1.3.14)**

Dieser Befehl setzt die Sollposition auf die Istposition. Dieser Befehl wir z.B. gebraucht, wenn beim löschen des Freeze der Motor die Bewegung nicht mehr zu Ende fahren soll, oder wenn der Motor stromlos war und nur bei Bestromung keine Bewegung machen darf.



2.4 Steuerung der einzelnen Zustände

Die einzelnen Zustände werden von der übergeordneten Steuerung (z.B. SPS) über vier digitale Signale (TRIG IN 1-4) aufgerufen. Jedem der 16 möglichen Eingangskombinationen wird ein Befehl zugeordnet. Diese Befehle erlauben es, Zustände direkt anzusteuern, den folgenden oder vorangehenden Zustand aufzurufen oder den selben Zustand nochmals auszuführen.

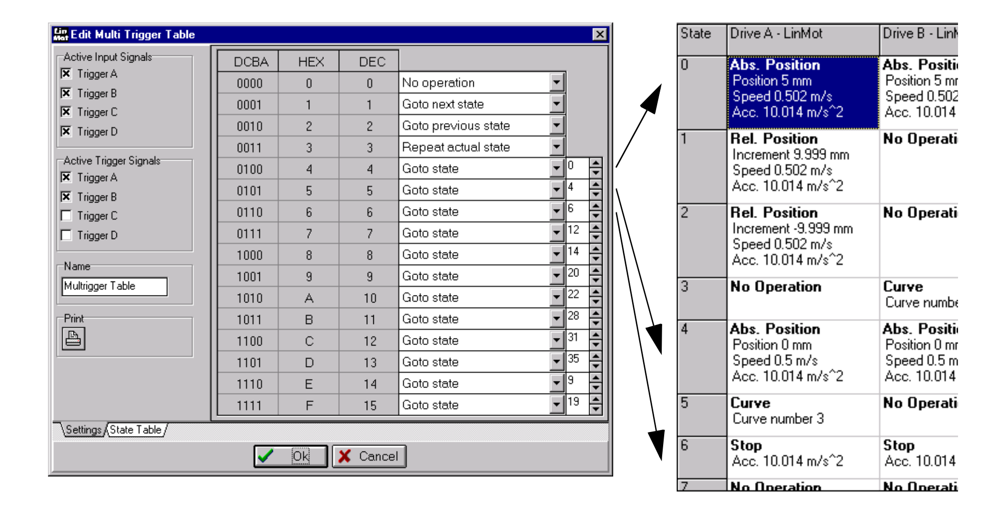


Abbildung 2-8: Ansteuerung der Zustände

No operation

Wechseln die Eingangssignale auf eine Eingangskombination für die der Befehl **No operation** definiert ist, wird der aktuelle Zustand beibehalten.

Falls die Bewegung des aktuellen Zustands noch nicht abgeschlossen ist, wird die angefangene Bewegung gemäss aktuellem Zustand beendet.

Goto next state

Wechseln die Eingangssignale auf eine Eingangskombination für die der Befehl **Goto next state** definiert ist, wird der dem aktuellen Zustand folgende Zustand ausgeführt.

In der Tabelle 2-1 auf Seite 23 ist ersichtlich, welche Bewegungen resultieren, wenn der Befehl **Goto next state** während einer angefangenen Bewegungen (des vorangehenden Zustands) aufgerufen wird.

Goto previous state

Wechseln die Eingangssignale auf eine Eingangskombination für die der Befehl **Goto previous state** definiert ist, wird der dem aktuellen Zustand vorausgehende Zustand ausgeführt.

Falls die Bewegung des aktuellen Zustands noch nicht abgeschlossen ist, können die resultierenden Bewegungen aus unten stehender Tabelle entnommen werden.

Angefangene Bewegung	Aufgerufene Bewegung	Resultierende Bewegung
Abs. Position A	No Operation	Abs. Position A
	Abs. Position B	Abs. Position B
	Rel. Position +B	Rel. Position A+B
	Curve 2	Curve 2

Tabelle 2-1: Resultierende Bewegung

Angefangene Bewegung	Aufgerufene Bewegung	Resultierende Bewegung
Rel. Position +A	No Operation	Rel. Position +A
	Abs. Position B	Abs. Position B
	Rel. Position +B	Rel. Position A+B
	Curve 2	Curve 2
Curve 1	No Operation	Curve 1
	Abs. Position B	Abs. Position B
	Rel. Position +B	letzter Sollwert von Curve 1 + B
	Curve 2	Curve 2
Stop	No Operation	Stop
	Abs. Position B	Abs. Position B
	Rel. Position +B	Rel. Position +B
	Curve 2	Curve 2
Freeze	No Operation	Freeze
	Abs. Position B	Freeze
	Rel. Position +B	Freeze
	Curve 2	Freeze

Tabelle 2-1: Resultierende Bewegung

Repeat actual state

Wechseln die Eingangssignale auf eine Eingangskombination für die der Befehl **Repeat actual state** definiert ist, wird der aktuelle Zustand nochmals ausgeführt. Soll derselbe Zustand mehrmals wiederholt werden, muss zwischen den **Repeat actual state** Befehlen jeweils die Eingangskombination für den **No Operation** Befehl vorgegeben werden, damit ein Wechsel der Eingangskombination und somit der Aufruf von **Repeat actual state** erfolgt.

Falls die Bewegung des aktuellen Zustands noch nicht abgeschlossen ist, können die resultierenden Bewegungen aus oben stehender Tabelle entnommen werden.

Goto state

Wechseln die Eingangssignale auf eine Eingangskombination für die der Befehl **Goto state** definiert ist, wird der betreffende Zustand ausgeführt.

Falls die Bewegung des aktuellen Zustands noch nicht abgeschlossen ist, können die resultierenden Bewegungen aus oben stehender Tabelle entnommen werden.

Aus dem in der folgenden Tabelle aufgeführten Beispiel geht hervor, wie die gewünschten Zustände mittels der vier digitalen Signale TRIG IN 1…4 von der übergeordneten Steuerung aufgerufen werden. Ein neuer Zustand wird erst ausgeführt, nachdem die Eingangskombination der Triggersignale



während einer einstellbaren Zeitdauer stabil anliegt (Jitterfilter). In der Tabelle sind die Bewegung von Motor A aus der oben stehenden Graphik aufgeführt.

Eingangs-signale	Befehl	Aktueller Zustand	Bewegung von Motor A
0100	Goto State 0	0	Positioniere auf abs. Position 5mm • max Verfahrgeschwindigkeit 0.5m/s • max Beschleunigung 10m/s^2
0001	Goto next state	1	Positioniere auf rel. Position +10mm • max Verfahrgeschwindigkeit 0.5m/s • max Beschleunigung 10m/s^2
0000	No Operation	1	-
0001	Goto next state	2	Positioniere auf rel. Position -10mm • max Verfahrgeschwindigkeit 0.5m/s • max Beschleunigung 10m/s^2
0000	No Operation	2	-
0011	Repeat actual state	2	Positioniere auf rel. Position -10mm • max Verfahrgeschwindigkeit 0.5m/s • max Beschleunigung 10m/s^2
0000	No Operation	2	-
0001	Goto next state	3	Motor bleibt auf aktueller Position bzw. die angefangene Bewegung wird abgeschlossen
0000	No Operation	3	-
0110	Goto State 6	6	Der Motor stoppt mit einer Beschleunigung von 10m/s^2
…	…	…	…

Tabelle 2-2: Abfolge einzelner Zustände

Betrieb im Analog Trigger Modus

Im Analog Trigger Modus verhalten sich die MT Servo Controller Ex00-MT und Ex000-MT wie die AT Servo Controller Ex00-AT und Ex000-AT. Bei dieser Einstellung kann nicht auf die in dieser Anleitung beschriebenen MT-Funktionen zurückgegriffen werden. Es stehen lediglich die im Benutzerhandbuch erläuterten AT-Funktionen zur Verfügung.



2.5 Konfigurations-Software

Die Multi Trigger Servo Controller Ex00-MT und Ex000-MT werden wie die Analog Trigger Servo Controller Ex00-AT und Ex000-AT mit Konfigurationssoftware *LinMot®* Talk konfiguriert. Für die Programmierung der MT Servo Controller wurde der **Curve Inspector** mit einer Eingabeoberfläche für die Multi Trigger Funktionen erweitert. Alle in der Bedienungsanleitung beschriebenen Funktionen der Konfigurationssoftware *LinMot®* Talk sind gleich geblieben und werden an dieser Stelle nicht näher erläutert.

Curve Inspector

In diesem Kapitel wird nur auf die Erweiterung des **Curve Inspector** mit der Multi Trigger Table eingegangen. Alle weiteren Funktionen des **Curve Inspector** sind im Bedienerhandbuch nachzulesen.

Neben den Kurven können mit dem **Curve Inspector** bei MT Servo Controllern auch **Multi Trigger**-Tabellen definiert werden. Diese werden wie normale Kurven verwaltet, können wie Kurven editiert, auf den Servo Controller geladen und vom Servo Controller gelesen werden. Falls in der Multi Trigger Tabelle Kurven aufgerufen werden, müssen diese zusammen mit der Multi Trigger Tabelle auf den Servo Controller geladen werden. Im Unterschied zu den Kurven kann auf einem Servo Controller nur eine Multi Trigger Tabelle gespeichert werden.

Multi Trigger Table

Falls der MT Servo Controller für den Multi Trigger Modus konfiguriert ist, steht im **Curve Inspector** neben der Taste **Create Curve** neu eine Taste **Create Multi Trigger Table** zur Verfügung.

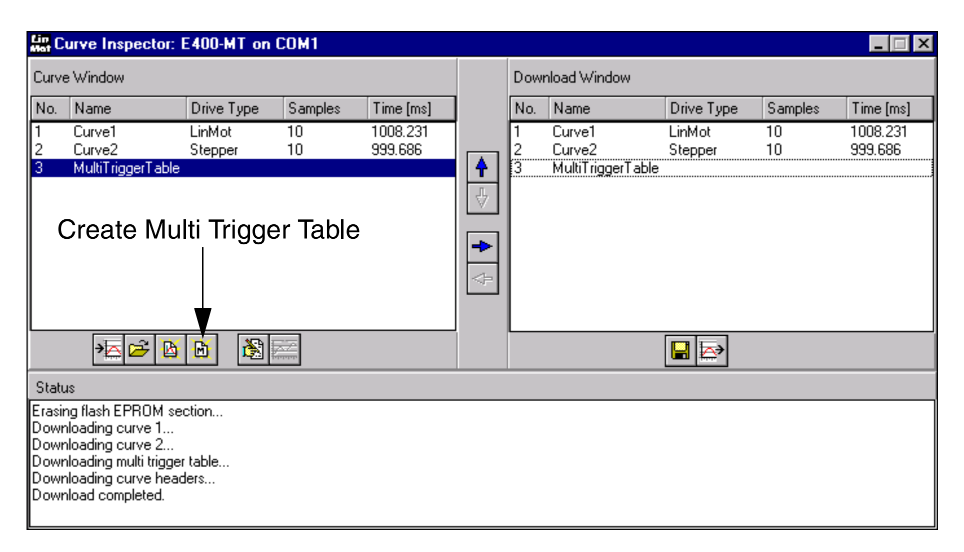


Abbildung 2-9: 'Create Multi Trigger Table'-Taste

Durch das Betätigen der Taste **Create Multi Trigger Table** öffnet sich das Dialogfenster **Edit Multi Trigger Table**, das eine einfache Programmierung des Servo Controllers erlaubt.

Das Dialogfenster **Edit Multi Trigger Table** besteht aus zwei Seiten, die mit den zwei Reitern **Settings** und **State Table** bezeichnet sind.

In der **State Table** werden die Sollpositionen und Sollwertkurven sowie die gewünschten Befehle für die einzelnen Motoren eingegeben. Die Zuordnung



der digitalen Eingangssignale bzw. Eingangskombinationen zu den einzelnen Zuständen wird auf der Seite **Settings** vorgenommen.

Bedienelemente in 'Settings'

Die erste Seite des Dialogfensters **Edit Multi Trigger Table** dient dazu, die aktiven Trigger zu setzen, die Namen von Multi Trigger Tabelle und **Input**-Tabelle zu definieren.

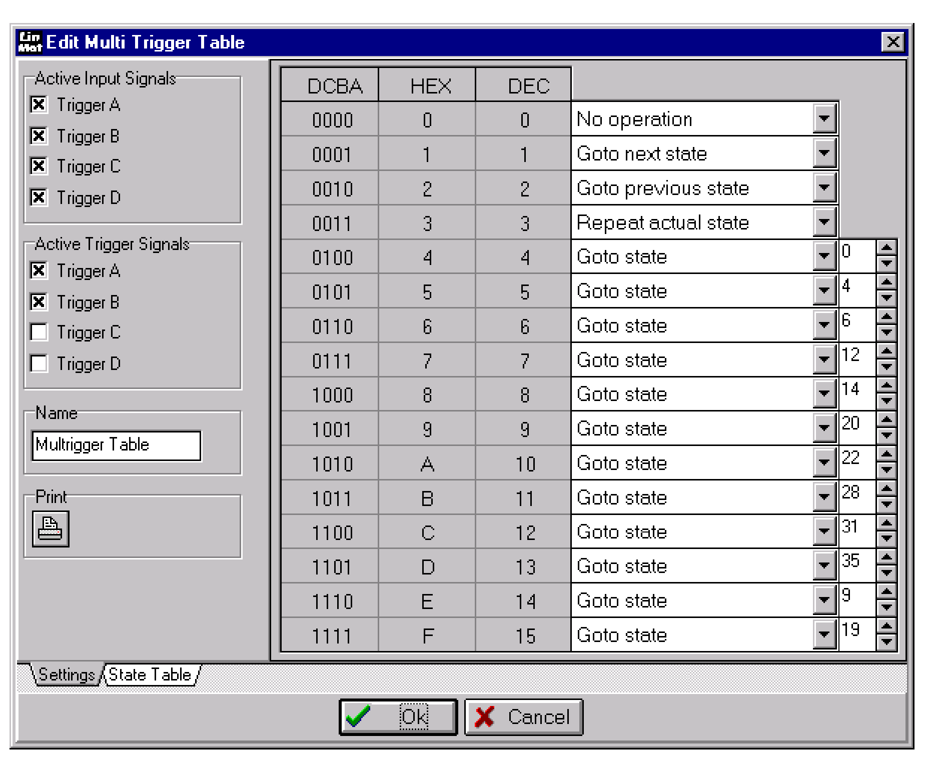
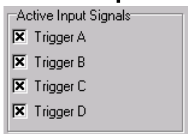


Abbildung 2-10: Settings

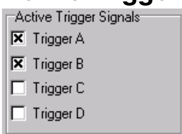
Active Input Signals



Durch die Felder **Active Input Signals** wird definiert, welche Eingangssignale von der übergeordneten Steuerung angesteuert werden. Sind alle vier Eingangssignale aktiviert, sind sechzehn Eingangskombinationen möglich (0-15), für die in Befehlstabelle ein zugehörendes Kommando definiert werden kann.

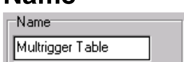
Achtung: Die aktivierten Triggersignale sind nur im Betriebszustand **RUN** aktiv. Werden die Triggereingänge zur Initialisierung benötigt, so müssen sie im Parameter Inspector unter **\System\IO Configuration** aktiviert werden.

Active Trigger Signals



Durch die Felder **Active Trigger Signals** werden diejenigen Eingangssignale angewählt, bei deren Pegelwechsel das entsprechende Kommando ausgeführt wird. Bei der Einstellung auf der Abbildung rechts, können die Pegel der Eingangssignale C und D Wechsel, ohne dass ein neuer Befehl ausgeführt wird. Erst bei einem Wechsel des Eingangssignals A bzw. B wird der für die aktuelle Eingangskombination definierte Befehl ausgeführt.

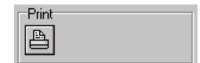
Name



Analog zu den Kurven kann auch der **Multi Trigger**-Tabelle ein Namen zugeordnet werden. Der gewünschte Name muss im Feld eingetragen werden.



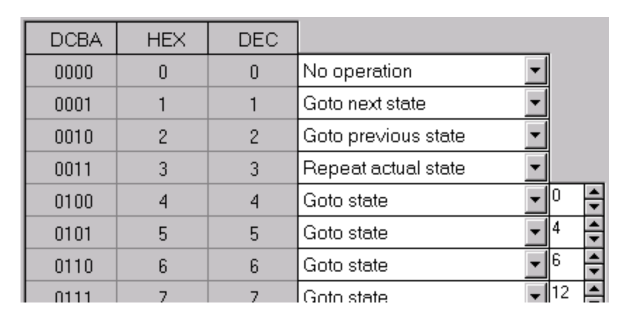
Print



Beim Betätigen diese Knopfes, wird die gesamte Konfiguration der **Multi Trigger**-Tabelle ausgedruckt.

Befehlstabelle

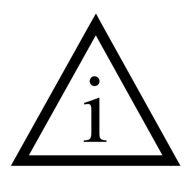
In der Befehlstabelle werden allen Kombinationen der Eingangssignale Befehle zugeordnet. Diese erlauben es, beispielsweise einen Zustand in der Zustandstabelle (siehe folgendes Kapitel) direkt aufzurufen, in den folgenden oder den vorhergehenden Zustand zu gehen, oder den aktuellen Zustand zu wiederholen.



DCBA	HEX	DEC		
0000	0	0	No operation	
0001	1	1	Goto next state	
0010	2	2	Goto previous state	
0011	3	3	Repeat actual state	
0100	4	4	Goto state	0
0101	5	5	Goto state	4
0110	6	6	Goto state	6
0111	7	7	Goto state	12

Abbildung 2-11: Befehlstabelle

In den ersten drei Spalten sind für alle Eingangskombinationen die Pegel der Eingangssignale in binärer, hexadezimaler und dezimaler Darstellung eingetragen. Über das Pull Down Menu der vierten Spalte lassen sich für die Eingangskombinationen die gewünschten Befehle einstellen. Die fünfte Spalte erscheint nur, wenn ein Zustand mit dem Befehl **Goto State** direkt aufgerufen wird. Dann lässt sich in der fünften Spalte die gewünschte Zustandsnummer (0-63) einstellen.



In der fünften Spalte lassen sich nur Zustandsnummern einstellen, wenn die betreffenden Zustände in der Zustandstabelle bereits definiert wurden.

In der Befehlstabelle können nur den Eingangskombinationen Befehle zugeordnet werden, für die alle benötigten Eingangssignale aktiviert sind. Eingangskombinationen, die nicht gültig sind, da die entsprechenden Eingangssignale nicht aktiviert sind, werden in der Befehlstabelle grau dargestellt.

Bedienelemente in “State Table”

In der Zustandstabelle **State Table** werden die Aktionen definiert, die jeder einzelne Aktor im betreffenden Zustand ausführen soll. Es können bis zu 64 Zustände gespeichert werden.

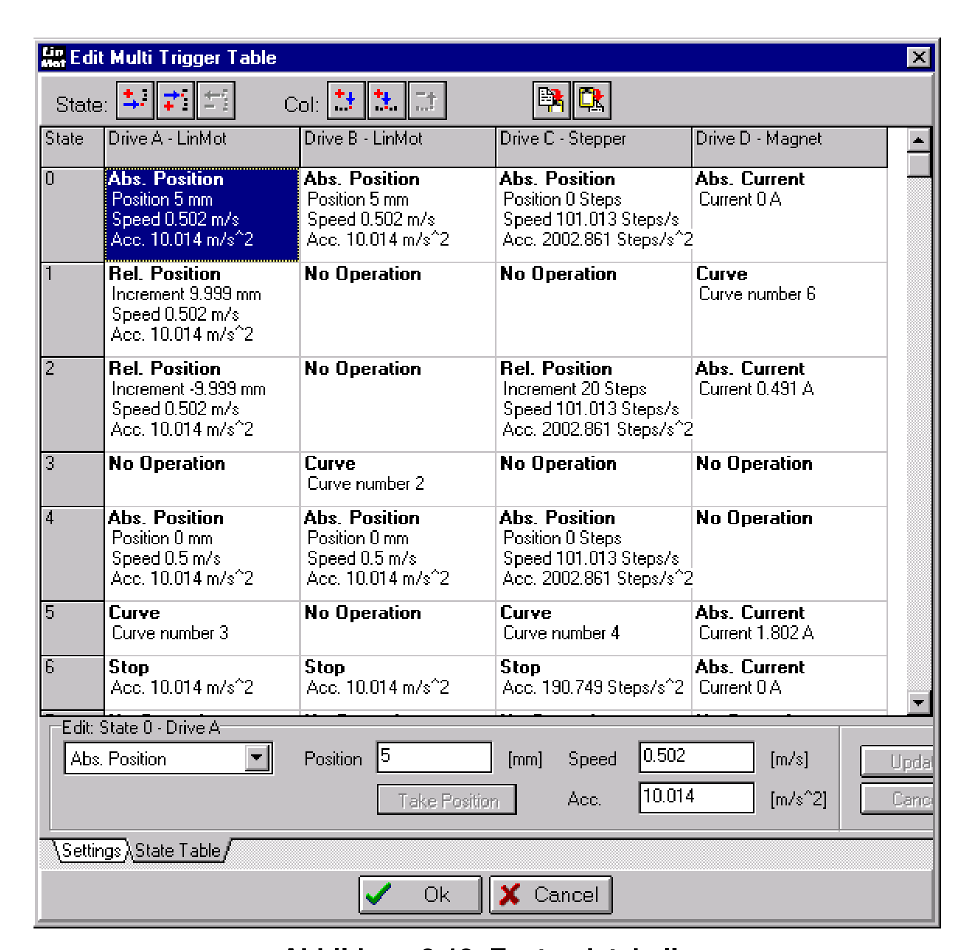
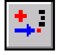


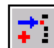
Abbildung 2-12: Zustandstabelle

Add State



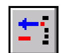
Ein neuer Zustand wird zuunterst an der Tabelle angehängt. Ein neuer Zustand kann nur dann eingefügt werden, wenn die Zustandstabelle weniger als 64 Zeilen aufweist.

Insert State



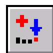
Ein neuer Zustand wird oben an den momentan selektierten Zustand eingeschoben. Ein neuer Zustand kann nur dann eingefügt werden, wenn die Zustandstabelle weniger als 64 Zeilen aufweist.

Delete State



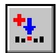
Die selektierte Zeile wird gelöscht. Diese Taste ist nur aktiv, wenn die ganze Zeile des betreffenden Zustand selektiert ist. Es können auch mehrere Zeilen aufs Mal gelöscht werden. Einmal gelöschte Zeilen könne nicht wieder hergestellt werden. Um ein unbeabsichtigtes Löschen zu verhindern, muss das Löschen von Zeilen bestätigt werden.

Add Column



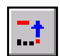
Eine neue Kolonne für einen weiteren Aktor wird auf der rechten Seite der Tabelle eingefügt. Es können mehr Aktoren eingefügt werden, als sich vom angeschlossenen Servo Controller ansteuern lassen. Dadurch kann ein Antrieb in einer neuen Kolonne definiert und später in eine andere Kolonne kopiert werden.

Insert Column



Eine neue Kolonne für einen weiteren Aktor wird auf der linken Seite der selektierten Kolonne eingefügt. Es können mehr Aktoren eingefügt werden, als sich vom angeschlossenen Servo Controller ansteuern lassen. Dadurch kann ein Antrieb in einer neuen Kolonne definiert und später in eine andere Kolonne kopiert werden.

Delete Column



Die selektierte Kolonne wird gelöscht. Diese Taste ist nur aktiv, wenn die ganze Kolonne selektiert ist. Es können auch mehrere Kolonnen aufs Mal gelöscht werden. Einmal gelöschte Kolonnen könne nicht wieder hergestellt werden. Um ein unbeabsichtigtes Löschen zu verhindern, muss das Löschen bestätigt werden.

Copy



Es können die Einträge von einzelnen Feldern, mehreren Feldern, ganzen Zeilen oder Kolonnen kopiert werden.

Paste



Die Einträge der kopierten Felder können mit dem Befehl **Paste** in andere Felder eingefügt werden. Der Befehl **Paste** funktioniert nur, wenn die Aktortypen vom kopierten Feld und vom einzufügenden Feld sowie die Anzahl Felder identisch sind.

Befehle

In den einzelnen Felder der Zustandstabelle werden die Bewegungen oder Befehle eingetragen, die der betreffende Aktor beim Aufruf des Zustandes ausführen soll. Die Zustandsbefehle und die dazugehörenden Einstellungen sind in Kapitel 4 beschrieben.

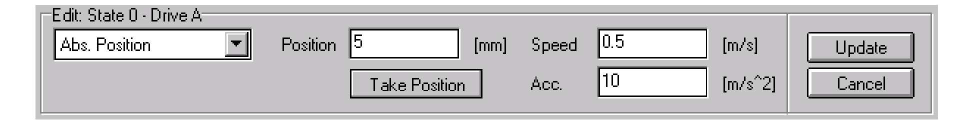


Abbildung 2-13: Eingabe der Zustands-Befehle

Der Befehl des selektierten Feldes der Tabelle wird mittels Pull Down Menu auf der linken Seite ausgewählt. Im Pull Down Menu befinden sich alle für den eingestellten Motortyp gültigen Befehle.

Die Felder rechts neben dem Pull Down Menu dienen zur Konfiguration des Befehls. Es werden jeweils nur die Felder angezeigt, die beim gewählten Befehl zur Verfügung stehen.

Mittels der Taste **Update** werden die vorgenommenen Einstellungen in das Feld übertragen. Falls die vorgenommenen Einstellungen nicht übernommen werden sollen, können die Änderungen durch Betätigen der Taste **Cancel** verworfen werden.

Save/Load



Mit der Taste **Ok** wird das **Edit Multi Trigger Table**-Fenster geschlossen und die Änderungen in der Multi Trigger Tabelle gespeichert. Mit der Taste **Cancel** kann das Fenster geschlossen werden, ohne dass die Änderungen gespeichert werden.

Nach dem Schliessen des Fensters befindet sich die erstellte oder editierte Tabelle im **Curve Window** des **Curve Inspector**. Die **Multi Trigger**-Tabelle kann nun wie eine Kurve ins **Download Window** verschoben und auf den Servo Controller geladen werden.



Set Port and Drive Type

Beim Doppelklick auf die erste Zelle von eine Spalte erscheint das Fenster mit dem Titel **Set Port and Drive Type**.

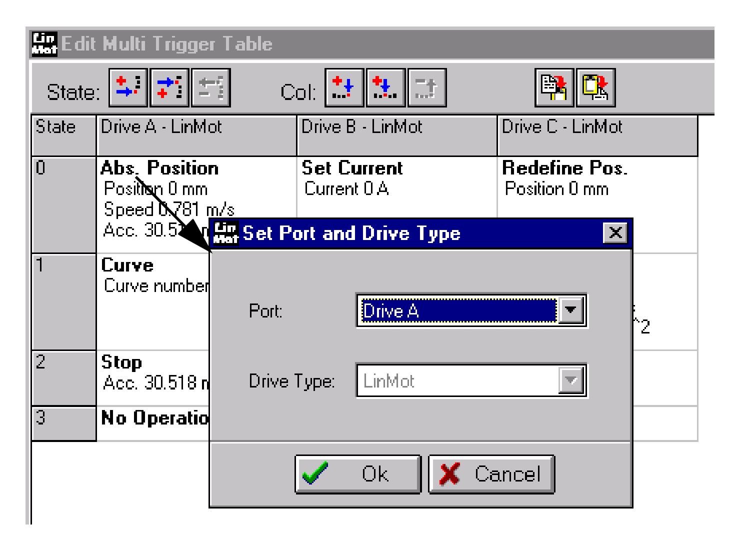


Abbildung 2-14: Fenster "Set Port and Drive Type"

In diesem Fenster kann der Port (Motorausgang auf dem Servo Controller) für die gewählte Spalte und Aktortyp geändert werden. Der Aktortyp lässt sich jedoch nur ändern, falls noch nichts editiert worden ist. Beim Drücken auf die Taste **Ok** werden die Änderungen gespeichert.

Es können auch Spalten ohne Zuordnung eines Motorausgangs definiert werden (Not Assigned). Dadurch können Spalten vertauschen werden, indem die betreffenden Spalten zuerst auf **Not Assigned** geändert werden und dann dem gewünschten Motorausgang zugeordnet werden.

Teach-In im Multi Trigger Modus

Die Programmierung der MT Controller der Serie Ex00-MT und Ex000-MT kann im Teach-In Verfahren vorgenommen werden, welches eine sehr schnelle und präzise Programmierung von Bewegungsabläufen erlaubt. Dabei wird der Läufer von Hand auf die gewünschte Sollposition gebracht und der aktuelle Positions-Istwert wird vom Servo Controller ausgelesen und in das gewünschte Feld in der Zustandstabelle eingetragen.

Die Teach-In Funktion steht beim Befehl **Abs. Position** in der Zustanstabelle zur Verfügung. Die aktuelle Läuferposition wird mittels Taste **Take Position** übernommen. Anschliessend muss definiert werden, mit welcher Geschwindigkeit und Beschleunigung die gespeicherte Position angefahren werden soll. Durch Betätigung der Taste **Update** werden die Werte in die Zelle der Zustandstabelle eingetragen.

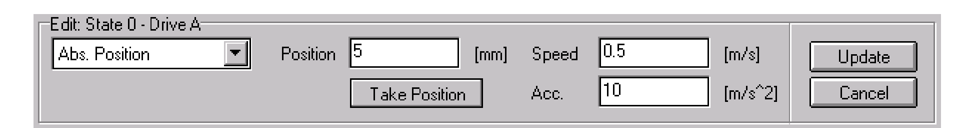


Abbildung 2-15: "Take Position" Taste für Teach-In

Für die Übernahme der Positions-Istwerte als Positions-Sollwerte muss der angeschlossene MT Servo Controller gestartet werden. Nach der Initialisierung werden die Motoren ausgeschaltet (durch Deaktivierung des RUN Eingangs). Jetzt können die Läufer auf die gewünschten Sollpositionen positioniert und die Werte in die Zustandstabelle übernommen werden.

3. PROFIBUS Servo Controller

Ab Release 1.3 werden die PROFIBUS-DP kompatiblen *LinMot®* Servo Controller unterstützt. Diese Controller verfügen über eine 12 MBit/s schnelle PROFIBUS-DP Schnittstelle und eignen sich hervorragend für komplexe Bewegungsabläufe in Verbindung mit SPS-Steuerungen, welche über eine integrierte PROFIBUS-DP-Master-Schnittstelle verfügen.

Die PROFIBUS-Beschreibung gliedert sich in die folgenden Kapitel:

- Kapitel 3.1 “Übersicht”
- Kapitel 3.2 “Zustandsautomat”
- Kapitel 3.3 “Verkabelung”
- Kapitel 3.4 “Inbetriebnahme ohne PROFIBUS”
- Kapitel 3.6 “Projektierung”
- Kapitel 3.7 “Datenmodul-Übersicht”
- Kapitel 3.8 “Datenmodule”
- Kapitel 3.9 “Diagnose”
- Kapitel 3.10 “Fehlersuche / Fehlerbehebung”
- Kapitel 3.11 “Schnittstellen”

3.1 Übersicht

offener Feldbus

PROFIBUS ist eine offene Feldbusnorm (EN 50170), die in der Automatisierungstechnik einen immer breiteren Anwenderkreis findet. Es gibt drei Ausprägungen von PROFIBUS: FMS, DP und PA. PROFIBUS-DP ist auf Geschwindigkeit optimiert und eignet sich deshalb speziell für die Kontrolle von hochdynamischen Antrieben wie *LinMot®* von einer übergeordneten Steuerung.

Master-Slave

PROFIBUS-DP arbeitet nach dem Master-Slave-Prinzip mit überlagertem Tokenpassing zwischen den verschiedenen Mastern. Die Master-Slave-Kommunikation ist streng zyklisch, wobei mit Hilfe einer Zeitüberwachung des Busses sowohl der Ausfall eines Masters, wie auch der Ausfall eines Slaves sofort erkannt wird. Zusätzlich ist die Diagnose eines Slaves durch einem Master standardisiert und bietet umfangreiche Möglichkeiten zur Übermittlung von Fehler- und Statusinformationen.

Die Identifikation eines PROFIBUS-DP Teilnehmers geschieht über eine einstellbare Adresse (0-125). Zusätzlich hat jede Geräteklasse eine sogenannte Identnummer, die für alle Geräte desselben Typs identisch ist (alle *LinMot®* DP Servo Controller besitzen dieselbe Identnummer).

www.profibus.com

In der folgenden Beschreibung wird davon ausgegangen, dass der Anwender ein Basiswissen über den PROFIBUS-DP besitzt. Eine sehr gute Informationsquelle über den PROFIBUS ist die WWW-Adresse **http://www.profibus.com**. Dort finden sich diverse Beschreibungen und weiterführende Literaturhinweise.



3.2 Zustandsautomat

Zum besseren Verständnis der Dokumentation sei im folgenden eine Kurzbeschreibung des Zustandsautomaten eines DP-Slaves wiedergegeben. Diese Beschreibung zeigt in welchen Zuständen sich ein DP-Slave befinden kann und welche Schritte er durchlaufen muss um ‘Online’ zu gehen. Mit ‘Online’ gehen ist der Zustand gemeint, in dem der Slave mit dem Master zyklisch Nutzdaten austauscht.

Was ist ein Slave?

Der *LinMot*® Servo Controller ist gemäss der PROFIBUS-Terminologie ein Slave. Er kann daher nicht von sich aus Datenübertragungen initieren, sondern muss immer vom sog. Master, meistens eine SPS, dazu aufgefordert werden.

Was ist ein Master der Klasse 1?

Ein Master der Klasse 1 führt Nutzdatenverkehr mit den ihm zugeteilten Slaves durch. Master der Klasse 1 sind meistens Industrie SPS-Steuerungen.

Was ist ein Master der Klasse 2?

Ein Master der Klasse 2 ist für Inbetriebnahmezwecke vorgesehen und kann kurzzeitig die Kontrolle über beliebige Slaves übernehmen. Master der Klasse 2 sind meistens PCs mit einer PROFIBUS-Karte.

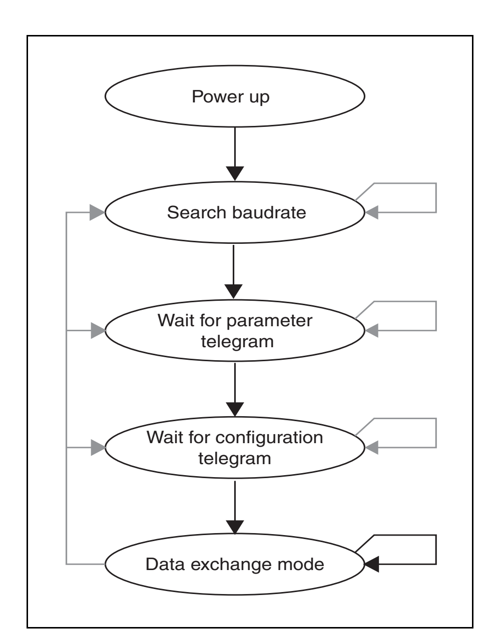


Abbildung 3-1: Zustandsmaschine von einem PROFIBUS-DP-Slave

Zustand	Beschreibung
Suche Baudrate	In diesem Zustand sucht der Slave die Baudrate mit der auf dem Bus kommuniziert wird.
Warte auf Parametriertelegramm	In diesem Zustand werden vom Slave nur Parametriertelegramme akzeptiert. Im Parametriertelegramm sind die von der Norm festgelegten Informationen wie z.B. PNO-Identnummer, Sync-Freeze-Fähigkeit usw. hinterlegt. *LinMot®* Servo Controller erwarten keine anwendungsspezifischen Parameterdaten.
Warte auf Konfigurationstelegramm	Das Konfigurationstelegramm legt die Anzahl und die Art der Ein- und Ausgangsdaten fest. *LinMot®* Servo Controller unterstützen verschiedene Datenmodule die beliebig zusammengestellt werden können. Es kann somit bei der Parametrierung festgelegt werden, welche Daten schliesslich im Datenübertragungsmodus gesendet werden. Ein Motor z.B. kann so konfiguriert werden, dass die Sollposition und die Istposition übertragen werden. Ein anderer Motor auf dem gleichen Servo Controller kann so konfiguriert werden, dass neben der Sollposition auch noch die maximale Geschwindigkeit übertragen wird.
Datenübertragunsmodus (auch Data_Exchange)	Wenn sowohl die Parametrierung als auch die Konfiguration von der Firmware des Slaves akzeptiert wurden geht der Slave in diesen Zustand über und tauscht zyklisch die konfigurierten Nutzdaten mit dem Master aus.

3.3 Verkabelung

In diesem Kapitel werden Hinweise und Vorschriften für eine korrekte Verkabelung des PROFIBUS-Netzwerkes gegeben.

Schirmung

Es sollten nur Kabel mit einem Geflechtschirm verwendet werden. Die Schirmung muss beidseitig grossflächig aufgelegt sein. Bei fest montierten Geräten ist es von Vorteil, wenn das geschirmte Kabel unterbrechungsfrei abisoliert und auf die Schirm- oder Schutzleiterschiene aufgelegt wird. Diese Massnahme erhöht die Betriebssicherheit bei stark gestörter Umgebung.

Busanschlusstecker

Es sollten nur Busanschlusstecker verwendet werden, die für PROFIBUS und die entsprechende Baudrate geeignet sind. Die Stecker an den beiden Enden des Busses sollten eine zuschaltbare Terminierung aufweisen. Diese Stecker gibt es z.B. von Siemens.

Potentialausgleich

Der Schirm des PROFIBUS-Kabels darf nicht zum Potentialausgleich verwendet werden. Bei Anlagen, die an verschiedenen Erdungspunkten geerdet sind, muss eine separate Erdungsleitung verlegt werden, deren Impedanz mindestens um den Faktor 10 kleiner ist als diejenige der Kabelabschirmung.



Terminierung

Bei Baudraten über 1.5 MBit/s muss der PROFIBUS an beiden Enden aktiv terminiert werden. Zusätzlich sollte in jedem Stecker für jede abgehende Datenleitung eine Längsinduktivitäten mit 100nH vorhanden sein.

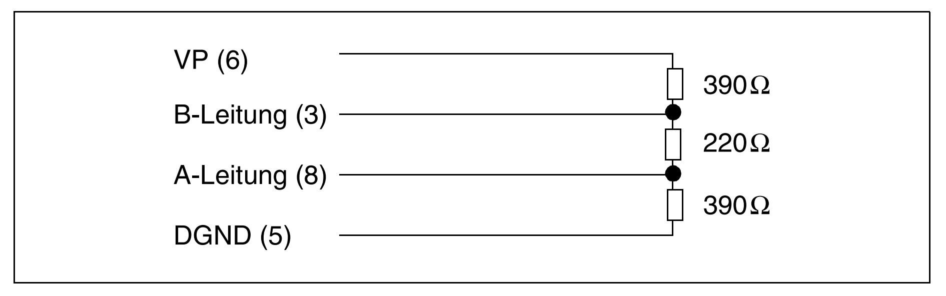


Abbildung 3-2: Leitungsabschluss der PROFIBUS-Leitung nach EN50170 (Pin-Nr. bei einem 9-poligen D-SUB Stecker)

Steckerbelegung

Die Steckerbelegung ist in der PROFIBUS-Norm festgeschrieben. Alle normkonformen Geräte müssen sich an diese Belegung halten. Die folgende Tabelle zeigt die Belegung der neunpoligen DSUB-Stecker.

Stift Nr.	Signal	Bedeutung
1	Schirm	Schirm / Schutzerde
2	M24	Masse der 24V Schutzerde
3	RxD/TxD-P[1]	Empfangs-/ Sendedaten -Plus, B-Leitung, rot
4	CNTR-P	Repeater Steuersignal (Richtungssteuerung)
5	DGND[1]	Datenübertragungspotential (Bezugspotential zu VP)
6	VP[1]	Versorgungsspannung-Plus, (P5V)
7	P24	Plus 24V Ausgangsspannung
8	RxD/TxD-N[1]	Empfangs-/ Sendedaten -N, A-Leitung, grün
9	CNTR-N	Repeater Steuersignal (Richtungssteuerung)

1) Diese Signale müssen vorhanden sein. Die anderen sind optional!

Bei komplizierter und weitverteilter Anordnung von PROFIBUS-Geräten an einem Busstrang empfiehlt sich ein gründliches Studium der technischen Richtlinie zum Aufbau von PROFIBUS-DP/FMS-Netzen. Dieses Handbuch mit der Bestell-Nr. 2.111 ist bei allen PROFIBUS-Nutzerorganisationen erhältlich. Die Adressen der PROFIBUS-Nutzerorganisationen gibt es auf dem Internet unter **http://www.profibus.com**.

3.4 Inbetriebnahme ohne PROFIBUS

Die PROFIBUS Servo Controller können in einem speziellen Modus, ohne eine funktionierende PROFIBUS-Verbindung, in Betrieb genommen werden. In diesem Modus kann z.B. der Regler eingestellt werden und die Stromversorgung überprüft werden. Im Inbetriebnahme-Modus wird ein Bewegungsprofil repetitiv abgefahren.

Mit den folgenden Schritten kann die Inbetriebnahme erfolgen:

1 PROFIBUS-Adresse auf 'EE' einstellen. Da die Adress Schalter nur beim Hochlaufen ausgelesen werden, ist bei jeder Änderung der Adresse ein Reset notwendig.
ACHTUNG: Wenn die Adresse auf "FF" gestellt wird, ist der Servo Controller im sog. Bootstrap Modus, bei dem keine LED aufleuchtet (Auch die Grüne nicht).

2 Mit der *LinMot*® Talk Software Bewegungsprofil erstellen und auf den Servo Controller herunterladen. Beim Bewegungsprofil sollte der Anfangspunkt dem Endpunkt entsprechen, da es ansonsten zu einer ruckartigen Bewegung kommt.

3 Motortyp-, Initialisierungs- und Reglerparameter einstellen. Siehe auch Kapitel "Parameter" auf Seite 61.

4 Parameter im Verzeichnis **\Drives\Drive X\Set Value Generation\Set Value Configuration** einstellen. Wenn der Parameter **Curve Number** auf 0 eingestellt wird, wird keine Kurve abgefahren!

5 Nun zuerst die Taste **Stop** und danach **Start** betätigen.

Wenn die Verkabelung korrekt gemacht wurde und alle Parameter richtig eingestellt worden sind, sollte der Motor nun das gewählte Bewegungsprofil zyklisch abfahren.

3.5 PROFIBUS Parameter

Die PROFIBUS spezifischen Parameter werden im Kapitel 6.9 beschrieben.

3.6 Projektierung

Die Projektierung der PROFIBUS-Slaves geschieht meistens mit Hilfe einer PC-Projektierungssoftware. Alle führenden SPS-Hersteller, wie u.a. Allen Bradley, Bosch, Mitsubishi, Omron und Siemens stellen eine solche Projektierungsumgebung zur Verfügung.

Gerätestammdatei

Die Basis für eine offene Projektierung bilden dabei die elektronischen Datenblätter, die in der PROFIBUS-Norm als Gerätestammdateien bezeichnet werden. Alle zur Projektierung des jeweiligen Slaves nötigen Informationen übernimmt die Projektierungssoftware aus diesen Gerätestammdateien. Durch die in der Norm festgelegte Definition der GSD-Dateien wird sichergestellt, dass jeder normkonforme Slave mit allen normkonformen Mastern zusammenarbeiten kann. Die GSD-Datei für die *LinMot*® Servo Controller hat den Namen **LINM00B6.GSD** und befindet sich nach der Installation der Software im Verzeichnis …\GSD der installierten *LinMot*® Talk Software (z.B. C:\LIN1R3R9\GSD).



Projektierung

Die eigentliche Projektierung geschieht meistens mittels **Drag and Drop**. Dabei werden typischerweise die folgenden Schritte durchlaufen:

1 Laden aller benötigten GSD-Dateien der verwendeten Slaves in die Projektierungssoftware. Dieser Schritt muss bei den gängigen Programmen nur ein einziges Mal durchgeführt werden.

2 Erstellen und Konfigurieren eines PROFIBUS-Bussystems am gewünschten Master.

3 Erzeugen der einzelnen Slaves am erstellten Bussystem.

4 Konfigurieren der einzelnen Slaves.

Beim Konfigurieren der Slaves können die gewünschten Datenmodule, welche im zyklischen Verkehr mit dem Master ausgetauscht werden, bestimmt werden. Alle von den *LinMot*® Servo Controllern unterstützten Datenmodule sind im Kapitel “Datenmodule” auf Seite 39 beschrieben.

In der Abbildung 3-3, “Projektierung von LinMot® PROFIBUS Servo Controllern” ist eine typische Projektierung eines PROFIBUS-Systems mit einem *LinMot*® Servo Controller gezeigt. Die gezeigte Projektierung wurde mit Hilfe der Siemens Step7-Software erstellt.

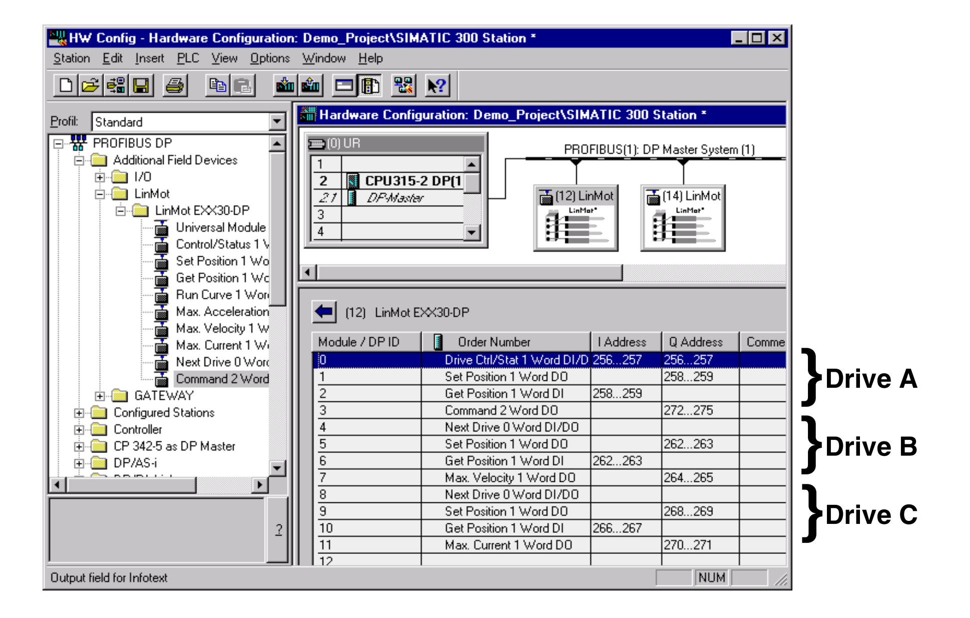


Abbildung 3-3: Projektierung von *LinMot*® PROFIBUS Servo Controllern

3.7 Datenmodul-Übersicht

Dieses Kapitel gibt einen Überblick über die Datenmodule, welche bei der Konfiguration von *LinMot*® PROFIBUS Servo Controllern benutzt werden können. Mit Hilfe dieser Module kann definiert werden, welche Daten zwischen dem PROFIBUS-Master und den *LinMot*® Servo Controllern ausgetauscht werden. Ein Modul kapselt jeweils einen untrennbaren Datenblock. Solch ein Block kann aus einem oder mehreren Datenwerten bestehen. Die folgende Tabelle gibt eine Übersicht über die vorhandenen Module.

Datenmodul	Beschreibung
Command	Führt ein Kommando auf dem *LinMot*® Servo Controller aus
Control/Status	Überträgt das Steuer- und Statuswort zum resp. vom Servo Controller
Get Position	Liest die aktuelle Position des Motors
Get Current	Liest den aktuellen Strom ein
Max. Acceleration	Setzt die maximale Beschleunigung
Max. Current	Setzt den maximalen Strom/Kraft
Max. Speed	Setzt die maximale Geschwindigkeit
Next Drive	Leitet einen neuen Motor ein
Run Curve	Startet ein abgelegtes Bewegungsprofil
Run Incr. Curve	Startet ein abgelegtes Bewegungsprofil, wobei die aktuelle Sollposition als Offset verwendet wird
Set Position	Setzt die Sollposition des Motors
Set Curve Speed	Setzt die Geschwindigkeit von Kurven
Set Curve Amplitude	Setzt die Amplitude von Kurven
Set Curve Offset	Setzt den Positionsoffset von Kurven

In der Projektierungssoftware tragen die Module nach dem eigentlichen Namen noch eine Kennung, welche die Richtung der Kommunikation sowie die Grösse der ausgetauschten Daten kennzeichnet.



3.8 Datenmodule

In der folgenden Beschreibung werden alle Datenmodule erklärt.

Command

Dieses Modul dient dazu, Kommandos an den *LinMot*® Servo Controller zu senden. Dieses Modul darf nur einmal pro Servo Controller projektiert werden. Da zur Laufzeit spezifiziert werden kann für welche Motoren der Befehl gilt, ist das nur eine geringe Einschränkung

Modul: Command	
Command ID	
Richtung	Master ➟Slave
Grösse	1 Wort
Aufbau	(siehe Tabelle unten)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Drive D	Drive C	Drive B	Drive A	Start Command (Toggle)	reserved	reserved	reserved	Bit 7 of Command ID	Bit 6 of Command ID	Bit 5 of Command ID	Bit 4 of Command ID	Bit 3 of Command ID	Bit 2 of Command ID	Bit 1 of Command ID	Bit 0 of Command ID

Command Value	
Richtung	Master ➟Slave
Grösse	1 Wort
Bereich	siehe Tabelle unten
Einheit	siehe Tabelle unten

Die Tabelle 3-1 auf Seite 40 beschreibt die Kommandos die mit diesem Modul ausgeführt werden können. Alle Kommandos werden durch das Ändern des Bits **Start Command** gestartet und gelten gleichzeitig für alle mit den Bits **Drive A…D** spezifizierten Motoren! Es ist daher möglich mit einem Kommando z.B. den **P**-Wert von allen Motoren zu verändern.

Die Kommandos **Redefine Position** und **Move Home Position** frieren die mit dem Modul **Set Position** gesetzte Position ein. Dies ist nötig um ungewollte Positionssprünge zu vermeiden. Mit dem Kommando **Unlock Set Position** wird das Positionsupdate wieder freigegeben. Die korrekte Benutzung dieser Kommandos geschieht daher in den folgenden Schritten:

1. Das Kommando **Redefine Position** ausführen. Mit diesem Befehl wird der aktuellen Position ein neuer Wert zugeordnet.
2. Die Sollposition, die mit dem Kommando Set Position gesendet wird, anpassen.
3. Das Kommado **Unlock Set Position** aufrufen um die Positionsvorgabe wieder freizugeben.

Kommando	**ID** **HEX**	**Wert** **von … bis**	**Einheit**	**Beschreibung**
No Command	0x00	-	-	Es wird kein Kommando ausgeführt
Redefine Position	0x01	-32256 … +32256	19.53125 µm oder 1/8 Step	Definiert die aktuelle Position neu und setzt die Sollposition auf den selben Wert. Dieser Befehl friert die mit dem Modul **Set Position** gesetzte Position ein. Sie kann mit dem Kommado **Unlock Set Position** wieder freigegeben werden.
Move Home Position	0x02	-32256 … +32256	19.53125 µm oder 1/8 Step	Verschiebt die Referenzposition um einen Wert. Die Sollposition wird so mitverschoben, dass sich der Motor nicht bewegt. Dieser Befehl friert die mit dem Modul **Set Position** gesetzte Position ein. Sie kann mit dem Kommado **Unlock Set Position** wieder freigegeben werden.
Unlock Set Position	0x03	-	-	Gibt die mit dem Modul **Set Position** gesetzte Sollposition frei. Dieser Befehl wird nur nach den Befehlen **Redefine Position** und **Move Home Position** benötigt.
Set Demand Position to Actual Position	0x04			Setzt die Sollposition auf die aktuelle Istposition des Motors
Set P	0x10	0 … 32640	0.00234 A/mm	Setzt den **P**-Wert des Reglers
Set I	0x11	0 … 32640	0.0457 A/(mm*s)	Setzt den **I**-Wert des Reglers
Set D	0x12	0 … 32640	0.015 A*s/mm	Setzt den **D**-Wert des Reglers
Set FF Friction	0x13	0 … 255	0.0234 A	Setzt den **FF Friction**-Wert des Reglers
Set FF Acceleration	0x14	0 … 32640	0.1 mA/(m/s[2)]	Setzt den **FF Acceleration**-Wert des Reglers
Set FF Deceleration	0x15	0 … 32640	0.1 mA/(m/s[2)]	Setzt den **FF Deceleration**-Wert des Reglers
Set Current Offset	0x16	-256 … 256	23.4 mA	Setzt den **Current Offset**-Wert des Reglers.

Tabelle 3-1: Übersicht über Kommando-Befehle des Command-Moduls



Control/Status

Das Datenmodul **Control/Status** übermittelt das Steuerwort zum *LinMot®* Servo Controller und liest das Statuswort vom Controllern zurück. Es muss als erstes Modul projektiert werden. Alle darauffolgenden Module beziehen sich auf den Motor A bis das Modul **Next Drive** folgt. Falls das Modul **Next Drive** von der Projektierungssoftware nicht unterstützt wird, kann alternativ wieder das Modul **Control/Status** zum Einleiten eines neuen Motors verwendet werden. Es werden aber nur die Daten im ersten **Control/Status**-Modul ausgewertet!

Modul: Control/Status	
Control	
Richtung	Master ➟Slave
Grösse	1 Wort
Aufbau	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FREEZE Drive D	FREEZE Drive C	FREEZE Drive B	FREEZE Drive A	Trig In 4 (Drive D)	Trig In 3 (Drive C)	Trig In 2 (Drive B)	Trig In 1 (Drive A)	reserved	reserved	reserved	FREEZE ALL Req.	INIT Request	STOP Request	RUN Request	reserved

Status	
Richtung	Slave ➟Master
Grösse	1 Wort
Aufbau	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	In Position Drive D	In Position Drive C	In Position Drive B	In Position Drive A	Curve Done D (Toggle)	Curve Done C (Toggle)	Curve Done B (Toggle)	Curve Done A (Toggle)	WARNING Pending	ERROR Pending	CMD Executed (Toggle)	DISABLE State	INIT State	ERROR State	RUN State	INIT Done

Das Steuerwort (Control) bestimmt in welchen Zustand der Servo Controller gehen soll und wird vom Master zum Slave gesendet. Die einzelnen Bits haben die folgende Bedeutung.

RUN Request	Fordert den Übergang in den Zustand **RUN** an
STOP Request	Fordert den Übergang in den Zustand **STOP** an
INIT Request	Fordert den Übergang in den Zustand **INIT** an
FREEZE ALL Req.	Fordert den Übergang in den Zustand **FREEZE** an. Dies Betrifft alle Motoren.

Diese Signale entsprechen denen der AT/MT Servo Controller. So wird z.B. durch Setzen von **INIT Request** der Übergang in den Initialisierungsmodus angefordert. Alle Zustände werden im Bedienerhandbuch im Kapitel 4.2 ab Seite 4-6 detailiert beschrieben.

Die Signale **Trig In 1** bis **Trig In 4** dienen der Initialisierung der Motoren in den Initialisierungsmodi **Trig Move Out** oder **Trig Move In**.

Trig In 1 (Drive A)	Trigger-Signal für Motor A
Trig In 2 (Drive B)	Trigger-Signal für Motor B
Trig In 3 (Drive C)	Trigger-Signal für Motor C
Trig In 4 (Drive D)	Trigger-Signal für Motor D

Mit den Signalen **FREEZE DRIVE A** bis **FREEZE DRIVE D** können die Bewegungen der Motoren selektiv unterbrochen werden. Die unterbrochenen Bewegungen werden weitergeführt, sobald das Signal zurückgesetzt wird..

FREEZE Drive A	FREEZE-Signal für Motor A
FREEZE Drive B	FREEZE-Signal für Motor B
FREEZE Drive C	FREEZE-Signal für Motor C
FREEZE Drive D	FREEZE-Signal für Motor D

Der Zustand in welchem sich der Servo Controller befindet wird durch das Statuswort zurückgeliefert. Dabei haben die einzelnen Bits die folgende Bedeutung:

RUN State	Servo Controller ist im Zustand **RUN**
ERROR State	Servo Controller ist im Zustand **ERROR**
INIT State	Servo Controller ist im Zustand **INIT**
DISABLE State	Servo Controller ist im Zustand **DISABLE**
INIT Done	Alle Motoren sind intialisiert

Das Bit **CMD Executed** ändert seinen Zustand bei jedem mit dem Modul **Command** ausgeführten Kommado.

CMD Executed	Dieses Bit ändert seinen Zustand mit jedem ausgeführten Kommando.

Die beiden Signale **ERROR Pending** und **WARNING Pending** zeigen an, ob ein Fehler oder eine Warnung anliegt.

ERROR Pending	Es liegt ein Fehler an
WARNING Pending	Es liegt eine Warnung an

Die vier Bits **Curve Done A..D** zeigen an, ob ein mit dem Modul **Run Curve** gestartetes Bewegungsprofil bereits beendet wurde. Die Bewegungsprofile werden durch das Wechseln (Togglen) eines Bits im Modul **Run Curve** gestartet. Sobald das Bewegungsprofil beendet ist, nimmt das Bit **Curve Done A..D** den gleichen Wert an, wie das Bit im Modul **Run Curve**. Mit dieser Methode kann jederzeit eindeutig bestimmt werden, ob ein Bewegungsprofil noch in Ausführung ist oder bereits beendet wurde.

Curve Done A	Bewegungsprofil beendet auf Motor A
Curve Done B	Bewegungsprofil beendet auf Motor B
Curve Done C	Bewegungsprofil beendet auf Motor C
Curve Done D	Bewegungsprofil beendet auf Motor D

Die vier Bits **In Position A..D** zeigen an, ob sich der Motor, nachdem eine neue Sollposition vorgegeben oder eine Kurve gestartet wurde, in einem eingegrenzten Bereich um den Zielpunkt der Bewegung befindet. Die Grenzen dieses Bereichs können mit den Parametern **In Position +** und **In Position -** im Verzeichnis **\Drives\Drive X\Position Monitoring** eingestellt werden.

In Position Drive A	Motor A befindet sich im Zielpunkt
In Position Drive B	Motor B befindet sich im Zielpunkt
In Position Drive C	Motor C befindet sich im Zielpunkt
In Position Drive D	Motor D befindet sich im Zielpunkt

Get Position

Dieses Modul dient dazu die aktuelle Position des Motors vom *LinMot*® Servo Controller auf die SPS zu übertragen. Wenn keine gültige Position vorhanden ist, da zum Beispiel noch nicht Initialisiert wurde, wird 0x7FFFh (32’767) übertragen.

<table>
<tr><th colspan="2">Modul: Get Position</th></tr>
<tr><td colspan="2">Actual Position</td></tr>
<tr><td>Richtung</td><td>Slave ➟Master</td></tr>
<tr><td>Grösse</td><td>1 Wort</td></tr>
<tr><td>Bereich</td><td>-32’256 bis +32’256; (32’767 bei ungültiger Position)</td></tr>
<tr><td>Einheit</td><td><table><tr><th>Motortyp</th><th>Einheit</th></tr><tr><td>LinMot®</td><td>19.53125 µm</td></tr><tr><td>Stepper</td><td>1/8 Step</td></tr><tr><td>Magnet</td><td>23.438 mA</td></tr></table></td></tr>
</table>

Get Current

Dieses Modul dient dazu den vom Regler eingestellten Sollstrom des Motors vom *LinMot*® Servo Controller auf die SPS zu übertragen.

<table>
<tr><th colspan="2">Modul: Get Current</th></tr>
<tr><td colspan="2">Actual Position</td></tr>
<tr><td>Richtung</td><td>Slave ➟Master</td></tr>
<tr><td>Grösse</td><td>1 Wort</td></tr>
<tr><td>Bereich</td><td>-256 bis +256</td></tr>
<tr><td>Einheit</td><td>23.438 mA</td></tr>
</table>

Max. Acceleration

Dieses Modul setzt die maximale Beschleunigung des Motors. Beim abfahren eines abgespeicherten Bewegungsprofils (Kurve) wird diese Beschleunigung nicht berücksichtigt. Wenn ein Wert ausserhalb des Bereichs vorgegeben wird, geht der Servo Controller in den Fehlerzustand (Data out of Range).

Modul: Max. Acceleration	
Max. Acceleration	
Richtung	Master ➟Slave
Grösse	1 Wort
Bereich	1 bis 1536
Einheit	

Motortyp	Einheit
LinMot®	238.419mm/s^2
Stepper	47.6836Steps/s^2
Magnet	-

Max. Speed

Dieses Modul setzt die maximale Geschwindigkeit des Motors. Beim Abfahren eines abgespeicherten Bewegungsprofils (Kurve) wird diese Geschwindigkeit nicht berücksichtigt. Wenn ein Wert ausserhalb des Bereichs vorgegeben wird, geht der Servo Controller in den Fehlerzustand (Data out of Range).

Modul: Max. Speed	
Max. Speed	
Richtung	Master ➟Slave
Grösse	1 Wort
Bereich	1 bis 24576
Einheit	

Motortyp	Einheit
LinMot®	190.735 µm/s
Stepper	0.081469Steps/s
Magnet	-

Max. Current

Dieses Modul setzt den maximalen Strom (Kraft) des Motors. Wenn dieses Modul verwendet wird, ist es wichtig, dass der Betriebswert vorgegeben wird, bevor das Initialisieren (Init Request) begonnen wird, da ansonsten der Anschlag nicht korrekt detektiert wird. Wenn ein Wert ausserhalb des Bereichs vorgegeben wird, geht der Servo Controller in den Fehlerzustand (Data out of Range).

Modul: Max. Current	
Max. Current	
Richtung	Master ➟Slave
Grösse	1 Wort
Bereich	0 bis 255
Einheit	23.438mA



Next Drive

Dieses Modul leitet den nächsten Motor bei der Projektierung ein. Einzelne Projektierungssoftwarepakete unterstützen dieses Modul nicht, da es keine Nutzdaten überträgt. Wenn dies der Fall ist kann stattdessen das Modul **Control/Status** verwendet werden.

Modul: Next Drive	
Next Drive	
Richtung	Master ➟Slave
Grösse	0 Wörter

Run Curve

Dieses Modul dient dazu, ein auf dem *LinMot*® Servo Controller abgelegtes, Bewegungsprofil zu starten. Mit jedem Signalwechsel des “Run Curve” Flags wird ein Profil gestartet.
Wenn das “**Continuous**” Flag gesetzt ist, wird das Profil kontinuierlich wiederholt.
Bei gesetztem “**Delayed**” Flag, wird das Profil erst aufgestartet, wenn das vorherige abgearbeitet ist. Dies erlaubt es, ein Profil aufzustarten, währenddem ein anderes noch läuft.
Bei gesetztemd “**Incremental**” Flag, wird die aktuelle Sollposition als “Curve Offset” verwendet. Bei der Verwendung von diesem Flag darf natürlich nicht gleichzeitit das “Curve Offset” Modul konfiguriert werden.
Sobald ein Profil gestarted wurde, wird die Sollpositionsvorgabe über ein allfällig konfiguriertes “Set Position” Modul gesperrt. Um das “Set Position” Modul wieder zu aktivieren, muss der Befehl “Unlock Set Position” im “Command” Modul verwendet werden .

Modul: Run Curve	
Curve	
Richtung	Master ➟Slave
Grösse	1 Wort
Aufbau	

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Run Curve (Toggle)	Continuous (ab Rel. 1.3.9)	Delayed (ab Rel. 1.3.9)	incremental (ab Rel. 1.3.9)	ME Mode (ab Rel. 1.3.10)*	reserved	reserved	reserved	reserved	reserved	Bit 5 of Curve Number	Bit 4 of Curve Number	Bit 3 of Curve Number	Bit 2 of Curve Number	Bit 1 of Curve Number	Bit 0 of Curve Number

**) Das “ME Mode” Flag ist nur gültig für die MP Software (Master Encoder mit Profibus). Wenn es gesetzt ist, wird die Kurve als Master Encoder Kurve gestartet.*

Run Incr. Curve

Dieses Modul sollte nicht weiter verwendet werden, da seine Funktionalität komplett über das “Incremental” Flag im “Run Curve” Modul integriert ist. Bestehende Konfigurationen, bei denen dieses Modul verwendet wird, funktionieren jedoch weiter, wie bisher.

Set Curve Speed

Dieses Modul dient dazu die gewünschte Geschwindigkeit von Kurven zu setzen. Wenn der maximale Wert gesetzt ist, wird die Kurve so schnell abgefahren wie sie erstellt wurde. Bei kleineren Werten geht die Geschwindigkeit linear zurück. Wenn ein Wert ausserhalb des Bereichs vorgegeben wird, geht der Servo Controller in den Fehlerzustand (Data out of Range)

Modul: Set Curve Speed	
Curve Speed	
Richtung	Master ➟Slave
Grösse	1 Wort
Bereich	0 bis 4096
Einheit	0.0244% der maximalen Geschwindigkeit

Set Curve Amplitude

Dieses Modul setzt die Kurven-Amplitude des Motors. Dabei entspricht der maximale Wert (4096) dem Skalierungsfaktor 100%. Bei diesem Faktor ist daher die Amplitude der Kurve so gross, wie sie im Kurveneditor erstellt wurde. Kleinere Werte verkleinern die Kurve entsprechend. Wenn ein Wert ausserhalb des Bereichs vorgegeben wird, geht der Servo Controller in den Fehlerzustand (Data out of Range)

Module: Set Curve Amplitude	
Curve Speed	
Richtung	Master ➟Slave
Grösse	1 Wort
Bereich	0 bis 4096
Einheit	0.0244% of maximum speed

Set Curve Offset

Dieses Modul wird dazu benutzt den Offset der Kurven einzustellen. Wenn ein Wert ausserhalb des Bereichs vorgegeben wird, geht der Servo Controller in den Fehlerzustand (Data out of Range).

Module: Set Curve Offset	
Curve Speed	
Richtung	Master ➟Slave
Grösse	1 Wort
Bereich	-32256 bis +32256
Einheit	(see table below)

Motor type	Unit
LinMot®	19.53125 µm
Stepper	1/8 Step
Solenoid	23.438 mA



Set Position

Dieses Modul dient dazu, die gewünschte Sollposition des Motors an den *LinMot*® Servo Controller zu übertragen. (Dieses Modul wird gesperrt, sobald eine Kurve aufgerufen wird. Um es wieder freizugeben, muss mit dem Command Modul ein “Unlock Set Position” durchgeführt werden). Wenn ein Wert ausserhalb des Bereichs vorgegeben wird, geht der Servo Controller in den Fehlerzustand (Data out of Range).

Modul: Set Position	
Demand Position	
Richtung	Master ➟ Slave
Grösse	1 Wort
Bereich	-32256 bis +32256
Einheit	

Motortyp	Einheit
LinMot®	19.53125 µm
Stepper	1/8 Step
Magnet	23.438 mA

3.9 Diagnose

Der PROFIBUS-DP ist für die Diagnose von Geräten mit Klartextmeldungen vorbereitet. Die *LinMot*® Servo Controller und die meisten Projektierungssoftwarepakete unterstützen diese. In den Diagnose-Meldungen erscheinen anliegende Warnungen und Fehler eines DP-Slaves im Klartext. Wenn die Diagnose mit einem SPS-Programm ausgewertet werden soll, kann auf die Beschreibung der Diagnose-Daten in der GSD-Datei zurückgegriffen werden. Dort ist die Bedeutung von allen Bits definiert.

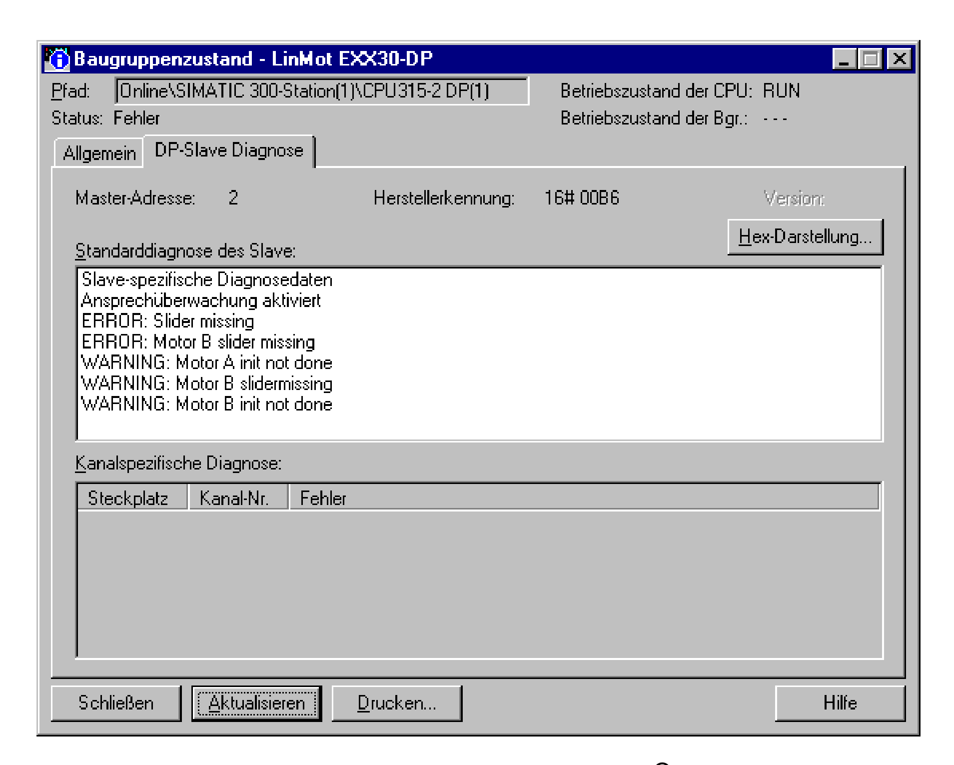


Abbildung 3-4: Diagnosemeldung eines *LinMot*® Servo Controllers

Das Diagnose Telegramm besteht aus 28 Byte. In den folgenden Tabellen wird die Zuordnung der Diagnosedaten zu Warnungen und Fehlern des LinMot Systems erläutert:

Byte Nr.	Bedeutung
0-5	Von der PROFIBUS Norm vorgegeben
6-7	Header und Padding
8-9	System Errors
10-11	Motor A Errors
12-13	Motor B Errors
14-15	Motor C Errors
16-17	Motor D Errors
18-19	System Warnings
20-21	Motor A Warnings
22-23	Motor B Warnings
24-25	Motor C Warnings
26-27	Motor D Warnings



System and Motor Errors		
Bit Nr.	**Fehler**	**Ursache**
0	Motor too hot calculated	Die berechnete Motor Temperatur ist zu hoch
1	Motor too hot sensor	Die gemessene Motor Temperatur ist zu hoch
2	Following Error	Schleppfehler
3	Slider Missing	- *LinMot*: Es ist kein Läufer im Motor, oder der Läufer ist zu weit ausgefahren. Ist der Läufer eventuell in der falschen Richtung eingebaut? - *Externe Sensorik:* Der Abstand des Sensorkopfes zum Band ist zu gross, oder das Band ist beschädigt.
4	Slave Error	Bei einem angeschlossener Booster oder Gantry Slave ist ein Fehler aufgetreten. Die Ursache ist in der Diagnose des Slave Motors ersichtlich.
5	Init failed	Beim Initialisieren ist ein Fehler aufgetreten
6	Motor Type mismatch	Der Typ des angeschlossenen Motors stimmt nicht mit der Konfiguration überein
7	Curve Missing	Es wurde versucht, eine nicht vorhandene Kurve zu starten
8	Reserved	
9	DCLV Power too low	Die Zwischenkreisspannung des Leistungsteils ist zu tief
10	DCLV Power too high	Die Zwischenkreisspannung des Leistungsteils ist zu hoch (z.B. durch Bremsen)
11	DCLVSignal too low	Die Spannungsversorgung für den Prozessorteil ist zu tief
12	DCLV Signal too high	Die Spannungsversorgung für den Prozessorteil ist zu hoch
13	Electronic Fault	E1000er Serie: Der Servo Kontroller ist zu heiss, oder die Kurz- und Erdschlussüberwachung hat angesprochen. E100er Serie: Der Servo Kontroller ist zu heiss.
14	Reserved	
15	Application Error	DP spezifischer Fehler (z.B data out of Range). Die genaue Ursache kann mit dem Error-Inspektor von LinMot Talk ermittelt werden.

System Warnings		
Bit Nr.	**Warnung**	**Ursache**
0	Motor too hot calculated	Die berechnete Motor Temperatur ist zu hoch
1	Motor too hot sensor	Die gemessene Motor Temperatur ist zu hoch
2	Following Error	Schleppfehler
3	Slider Missing	- *LinMot*: Es ist kein Läufer im Motor, oder der Läufer ist zu weit ausgefahren - *Externe Sensorik*: Der Abstand des Sensorkopfes zum Band ist zu gross, oder das Magnetband ist beschädigt.
4	Reserved	
5	Reserved	
6	Init not done	Die Initialisierung ist noch nicht abgeschlossen
7	Reserved	
8	Low Free Capacity	Knappe Systemressourcen
9	DCLV Power small	Die Zwischenkreisspannung des Leistungsteils ist tief
10	DCLV Power high	Die Zwischenkreisspannung des Leistungsteils ist hoch (z.B. durch Bremsen)
11	DCLVSignal small	Die Spannungsversorgung für den Prozessorteil ist tief
12	DCLV Signal high	Die Spannungsversorgung für den Prozessorteil ist hoch
13	Electronic Fault	Der Servo Controller ist heiss
14	Emergency Stop	Der Servo Controller befindet sich im Notstop-Zustand.
15	Reserved	



Motor Warnings		
Bit Nr.	**Warnung**	**Ursache**
0	Motor hot calculated	Die berechnete Motor Temperatur ist hoch
1	Motor hot sensor	Die gemessene Motor Temperatur ist hoch
2	Following Error	Schleppfehler
3	Slider Missing	- LinMot: Es ist kein Läufer im Motor, oder der Läufer ist zu weit ausgefahren. - Externe Sensorik: Der Abstand des Sensorkopfes zum Band ist zu gross, oder das Band ist beschädigt.
4	Slave Warning	Bei einem angeschlossener Booster oder Gantry Slave ist eine Warnung aufgetreten. Die Ursache ist in der Diagnose des Slave Motors ersichtlich.
5	Reserved	
6	Init not done	Die Initialisierung ist noch nicht abgeschlossen
7	Reserved	
8	Mot not in Pos Range	Der Motor ist nicht innerhalb der PosRange (in LinTalk konfiguriert)
9	DCLV Power small	Die Zwischenkreisspannung des Leistungsteils ist tief
10	DCLV Power high	Die Zwischenkreisspannung des Leistungsteils ist hoch (z.B. durch Bremsen)
11	DCLVSignal small	Die Spannungsversorgung für den Prozessorteil ist tief
12	DCLV Signal high	Die Spannungsversorgung für den Prozessorteil ist hoch
13	Electronic Fault	Der Servo Controller ist heiss
14	Reserved	
15	Reserved	

3.10 Fehlersuche / Fehlerbehebung

Der PROFIBUS-DP ist ein sehr robuster Industriebus. Probleme bei der Inbetriebnahme sind daher meistens auf eine fehlerhafte Verkabelung oder Parametrierung und nicht auf defekte Geräten zurückzuführen. Bei Kommunikations-Problemen sollte in jedem Fall zuerst die Busverkabelung und danach die Parametrierung überprüft werden. Die folgenden Tips haben sich in der Praxis bewährt:

SPS urlöschen

Es empfiehlt sich die SPS jedesmal zuerst komplett urzulöschen, wenn eine Änderung an der Konfiguration oder der Software vorgenommen wurde.

Servo Controller geht nicht in den Online-Zustand

DP-Adresse überprüfen: Die Adresse wird mittels der beiden Drehschalter auf der Frontseite des Servo Controller eingestellt. Achtung: Die Adresse wird hexadezimal eingegeben!

Verkabelung überprüfen. Es dürfen nur die zwei Stecker am Busende terminiert werden. Da der Bus aktiv terminiert wird, müssen zur korrekten Terminierung die beiden Geräte am Ende des Busses eingeschaltet sein, damit die Stecker mit Spannung für die Terminierung versorgt werden. Es ist allenfalls zu überprüfen, ob die entsprechenden Geräte auch eine Speisung für die externe Terminierung zur Verfügung stellen. Im Zweifelsfall an einem Ende einen *LinMot*® Servo Controller und am anderen Ende die SPS plazieren.

Bei 12MBit/s sind keine Stichleitungen mehr zuglelassen.

Version der GSD-Datei überprüfen.

Motor initialisiert nicht

Falls Datenmodule für maximale Beschleunigung, maximale Geschwindigkeit oder maximalen Strom konfiguriert wurden, müssen diese mit geeigneten Werten vorbelegt werden, damit der Motor initialisieren kann.

Byte-Reihenfolge

Leider verwenden die verschiedenen SPS-Hersteller in ihren Geräten verschiedene Definitionen der Byte-Reihenfolge. Bei einigen Herstellern hat das tiefstwertige Byte (LSB) in einem Wort die kleinste Adresse, bei anderen verhält es sich genau umgekehrt. Um dieses Problem zu umgehen kann beim *LinMot*® Servo Controller die Byte-Reihenfolge mittels Parameter eingestellt werden. Eine falsch konfigurierte Byte-Reihenfolge führt dazu, dass in einem Wort das untere mit dem oberen Byte vertauscht wird.

hohe Zykluszeit

Bei leistungsschwachen SPS-Steuerungen empfiehlt es sich die hochpriorisierte Diagnose auf den *LinMot*® Servo Controllern auszuschalten, um die maximale Zykluszeit[1] zu verkleinern. Im Fehlerfall wird somit kein hochpriorisierter Interrupt auf der SPS ausgelöst. Die Fehlerbehandlung muss dann im zyklischen OB geschehen.

Um eine kleine Zykluszeit zu erreichen empfiehlt es sich, die *LinMot*® Servo Controller in das Prozessabbild einzubinden und nicht mit Peripheriezugriffen darauf zuzugreifen.

Je nach Anwendung und Steuerung kann es sinnvoll sein, wenn die *LinMot*® Servo Controller in ein eigenes Teilprozessabbild eingebunden werden. Dadurch entfällt die zyklische Prozessabbildaktualisierung.

1) Es ist hier die Zykluszeit auf der SPS gemeint. Die Buszykluszeit auf dem PROFIBUS ist in den meisten Fällen von der Zykluszeit der SPS unabhängig.



3.11 Schnittstellen

Die PROFIBUS Servo Controller haben die folgenden Schnittstellen:

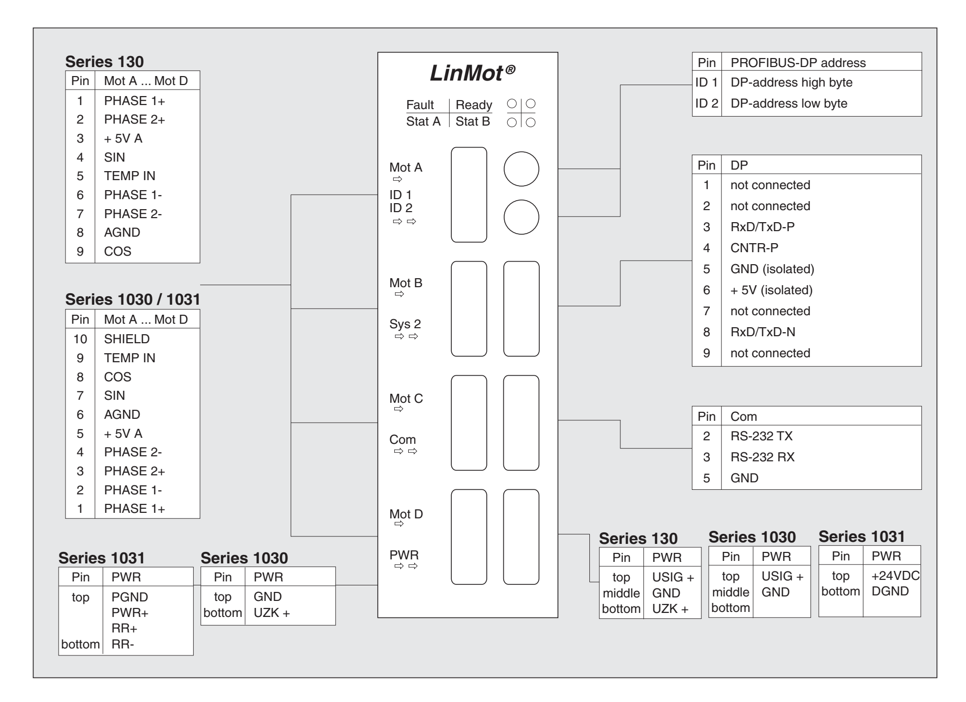


Abbildung 3-5: Pinbelegung der PROFIBUS Controller

4. Externe Positionssensorik

Die Wiederholgenauigkeit bei den Linearmotoren der Serie *LinMot®* P mit integriertem Messsystem kann mit einer zusätzlichen externen Positionssensorik bis auf 10µm erhöht werden. Die hohe Linearität des externen Messbandes ermöglicht den Einsatz der Linearmotoren in Applikationen, die eine grosse absolute Positioniergenauigkeit erfordern.

LinMot® Servo-Controller unterstützen zwei Sensortypen: Sinus/Cosinus und A/B (inkrementell).

4.1 Sinus/Cosinus Sensor

Funktionsprinzip

Die Positionsgeber bestehen aus einem Abtastkopf und einem Massband. Wenn der Abtastkopf über das Massband fährt, wird am Ausgang ein Sinus- und ein Cosinussignal ausgegeben. Aus diesen Signalen berechnet der *LinMot®* Servo Controller ein Positionssignal, welches für die Positionierung der Motoren genutzt wird.

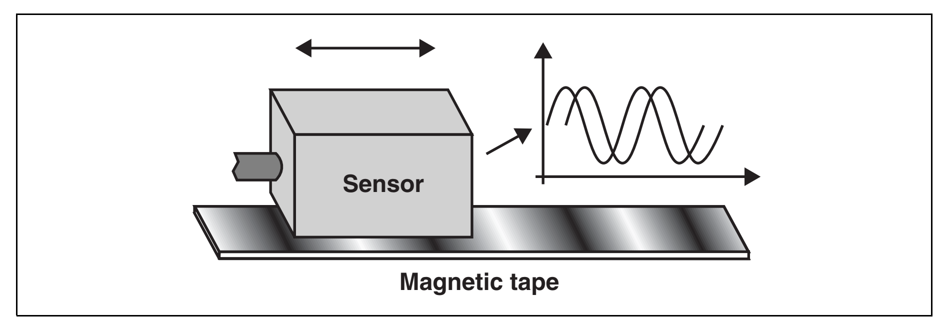


Abbildung 4-1: Funktionsprinzip Positionssensorik

Anschluss der Sensorik

Die externe Sensorik wird wie ein Linearmotor an einem Motorkanal angeschlossen. Zur Pegelanpassung muss dedoch eine “Sensorverstärkerbox” SA01 vorgeschaltet werden. Der korrespondierende Motor, der die externe Sensorik nutzen soll, muss nun an den darauffolgenden Motorkanal angeschlossen werden.

Beispiel: Wird die Sensorik an Motorkanal A angeschlossen, muss der Motor an Motorkanal B angeschlossen werden. Wird die Sensorik an Motorkanal C angeschlossen, muss der Motor an den Kanal D angeschlossen werden.

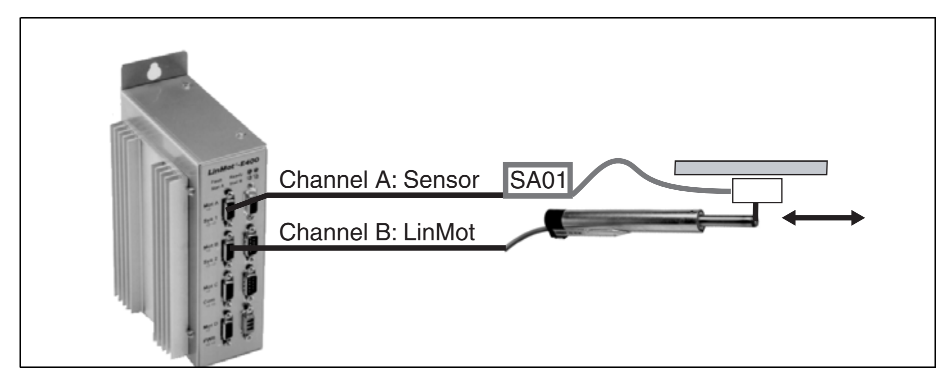


Abbildung 4-2: Anschluss einer externer Positionssensorik



Betrieb mit *LinMot*® Positionssensoren MS01-1000/P und MS01-5000/P

Bei einem Betrieb mit den *LinMot*® Positionssensoren MS01-1000/P oder MS01-5000/P wird der Sensorverstärker SA01-44/D benötigt. Dieser übernimmt die Verstärkung der differentiellen Sinus- und Cosinus-Signale des Sensors. Die nachfolgende Abbildung zeigt, wie die Sensorik angeschlossen werden muss.

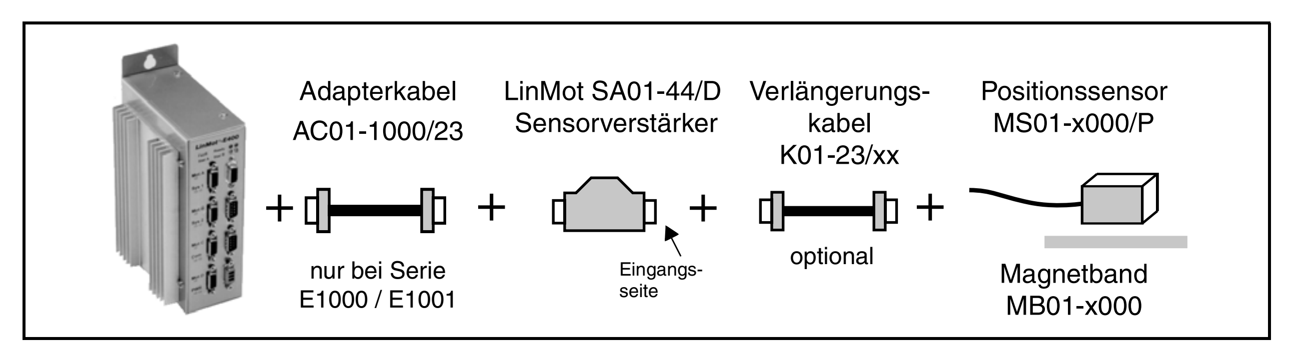


Abbildung 4-3: Anschluss einer *LinMot*® Positionssensorik mit Sensorverstärker SA01-44/D

Die folgende Tabelle zeigt die Pinbelegung auf der Eingangseite des Sensorverstärkers. Sowohl Sinus- wie auch Cosinussignal haben aus Gründen der elektromagnetischen Störsicherheit differentielle Eingänge .

Signalname	*LinMot*® Sensorverstärker	Beschreibung
GND	Pin 8	Masse
+5V	Pin 7 (Ausgang)	+5V Speisung
Sensor Sin-	Pin 2 (Eingang)	Diff. Sinus Eingang Amplitude: ±50mV Common Mode Bereich: 0…5V
Sensor Sin+	Pin 1 (Eingang)	
Sensor Cos-	Pin 5 (Eingang	Diff. Cosinus Eingang Amplitude: ±50mV Common Mode Bereich: 0…5V
Sensor Cos+	Pin 4 (Eingang)	

Betrieb mit Positionssensoren von Fremdherstellern

Der Betrieb mit analogen Sinus-/Cosinus-Positionssensoren von Fremdherstellern ist grundsätzlich möglich, wird jedoch nicht empfohlen.

Positionssensoren von Fremdherstellern müssen Sinus- und Cosinus-Signale mit einer Offsetspannung von +2.5V und eine Amplitude von maximal ±2.3V aufweisen. Für den Stecker gilt die folgende Pinbelegung:

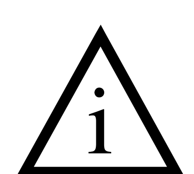
Signalname	*LinMot*® E100 Serie	*LinMot*® E1000/ E1001 Serie	Beschreibung
GND	Pin 8	Pin 6	Masse
$+5V_{out}$	Pin 3	Pin 5	5 Volt Ausgang Max. 50 mA
Sensor Sin	Pin 4	Pin 7	Sinus Eingang Offset: $+5V_{out}$ / 2 Amplitude: ±2.3 Volt
Sensor Cos	Pin 9	Pin 8	Cosinus Eingang Offset: $+5V_{out}$ / 2 Amplitude: ±2.3 Volt

Auflösung

Die folgende Tabelle zeigt, wie die Auflösung mit dem Hubbereich der *LinMot*® Motoren zusammenhängt.

Auflösung und Hubbereich		
Polteilung	**Auflösung**	**Hubbereich**
1mm	20µm	1260mm
	10µm	630mm
	5µm	315mm
	2.5µm	157.5mm
	1.25µm	78.75mm
5mm	20µm	1260mm
	10µm	630mm
	5µm	315mm

LinMot® Talk



Wenn eine externe Positionssensorik benutzt wird und eine Auflösung ungleich ’20µm’ eingestellt ist, muss dies bei den folgenden Parametern berücksichtigt werden:

- Home/Check Init/Initial Position
- Minimal/Maximal Position
- 0V/10V/’0’/’1’ Posititon
- alle Parameter im Verzeichnis ’Position Monitoring’
- alle Parameter (Position, Geschwindigkeit und Beschleunigung) bei der Erzeugung von Bewegungsprofilen im ‘Curve Editor’.
- alle Positionen (nicht aber Geschwindigkeiten und Beschleunigungen) bei Einsatz von: Mutitriggertabellen, dem ASCII-Protokoll und dem PROFIBUS.

Alle diese Parameter müssen gemäss der folgenden Formel umgerechnet werden:

$$\mathbf{Wert_{neu} = Wert_{real} * Faktor}$$

Dabei hat der Faktor den folgenden Wert:

Auflösung	Faktor
10µm	2
5µm	4
2.5µm	8
1.25µm	16

Beispiel

Die Auflösung ist auf ‘5µm’ eingestellt. Wenn nun der Parameter **Home Position** auf den Wert 30mm eingestellt werden soll, muss stattdessen ein Wert von 120mm eingetragen werden.



4.2 A/B sensors

Funktionsprinzip

Die A/B Positionssensorik besteht aus einem Sensorkopf und einem Magnetband oder einem Massband. Es werden zwei digitale Ausgangssignale A und B generiert, welche um 90 Grad gegeneinander phasenverschoben sind. Die Signale A und B werden an einen Encodereingang der Masterencoder-Schnittstelle mittels RS-422 Pegel geführt. Aus diesen Signalen generiert der *LinMot*® Servocontroller die Position, welche für die Regelung von Motoren benutzt werden kann.

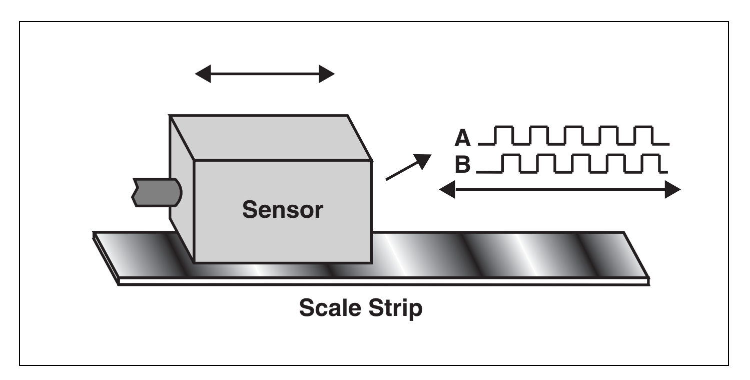


Abbildung 4-4: Funktionsprinzip des A/B Positionssensors

Anschluss der Sensorik

Der externe A/B-Sensor wird an einem Encoder-Stecker auf der Masterencoder-Schnittstelle angeschlossen. Siehe “Addendum Master Encoder”. Die Signale sind RS422 und der Encoder kann über die Software frei einem Motor zugeordnet werden.

Für die Masterencoder-Funktionalität werden bei den Version 2 Kontrollern Zusatzmodule gebraucht, wlche folgende Schnittstellen abieten:

	ME01-01/08	ME01-02/08
Link A	**ME / Pos Sensor**	**ME / Pos Sensor**
Link B	**nur loop through**	**Pos Sensor**

Bei Version 3 Kontroller ist mit der Option -ME die Masterencoder-Funktionalität bereits unterstützt, es braucht keine Zusatzmodule mehr.



Parameter

Der einzige Parameter, welcher zu setzen ist, damit der Motor seine aktuelle Position über die A/B-Sensorik bekommt, ist unter Drive X\Advanced\Position Sensor. Die Auflösung des Encoders kann mit der Dekodierung ausgewählt werden. Es gibt folgende Möglichkeiten:

Spezifische Punkte für A/B-Sensoren in \Drives\Drive X\Advanced\Position Sensor	
AB Enc1 1X	Position von Encoder Link 1, Decode Modus 1x
AB Enc1 2X	Position von Encoder Link 1, Decode Modus 2x
AB Enc1 4X	Position von Encoder Link 1, Decode Modus 4x
AB Enc2 1X	Position von Encoder Link 2, Decode Modus 1x
AB Enc2 2X	Position von Encoder Link 2, Decode Modus 2x
AB Enc2 1X	Position von Encoder Link 2, Decode Modus 4x



5. Master/Slave-Betrieb

Auf einem *LinMot*® Controller können bis zu vier Motoren zusammen als eine logische Achse betrieben werden. Zwei unterschiedliche Master/Slave Modi sind unterstützt: Master/Booster und Master/Gantry Slave.

5.1 Master/Booster Betrieb

Der Master/Booster-Betrieb gestattet es, die verfügbare Kraft für eine Bewegung zu erhöhen, indem Motoren parallel geschaltet werden. Dazu kann ein Motor als Master und bis zu drei Motoren als Booster definiert werden. Im Master/Booster Betrieb müssen die Läufer der Motoren mechanisch aneinander gekoppelt werden.

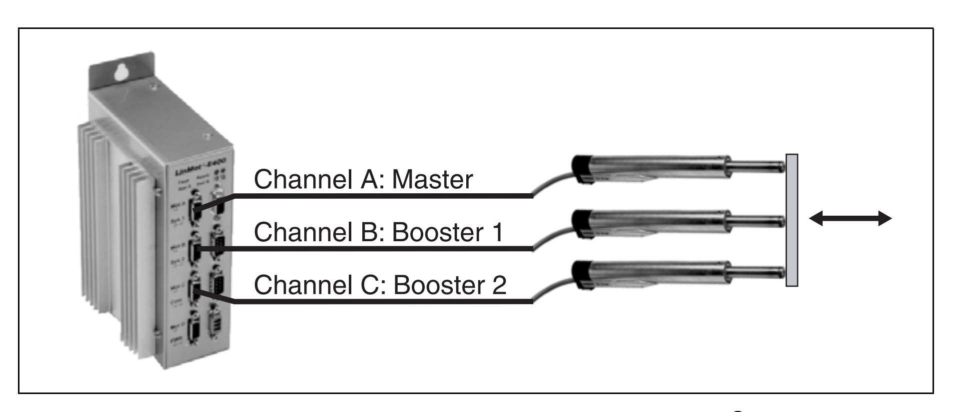


Abbildung 5-1: Master/Booster-Betrieb von *LinMot*® Motoren

Funktionsprinzip

Beim Betrieb wird dem Master-Motor die Position vorgegeben. Der vom Positionsregler auf dem Master-Motor erechnete Sollstrom wird nun auf den Master- sowie die Booster-Motoren vorgegeben.

Parameter

Bei der Parametrierung der Motoren muss der Master-Motor vor den Booster-Motoren definiert werden! Bei Booster-Motoren müssen nur noch die folgenden Parameter definiert werden:

- Motortyp (P0x-23, P0x-37, P0x-48), muss gleicher Typ sein wie der Master
- Einstellung 2 oder 3 Ampère (nur bei E100 Serie), gleicher Wert wie Master
- Kommutierung
- Fehlerbehandlung



Betriebsarten

Ein Booster-Motor kann entweder in Richtung des Masters oder entgegengesetzt zum Master Kraft ausüben. Dies kann im Verzeichnis **Master / Booster** festgelegt werden. Bei der Einstellung **Booster parallel** muss der Booster-Motor in Richtung des Masters zeigen, bei der Einstellung **Booster reverse** muss der Booster in Gegenrichtung des Masters montiert werden. Siehe auch Abbildung 5-2, “Booster-Betriebsarten”.

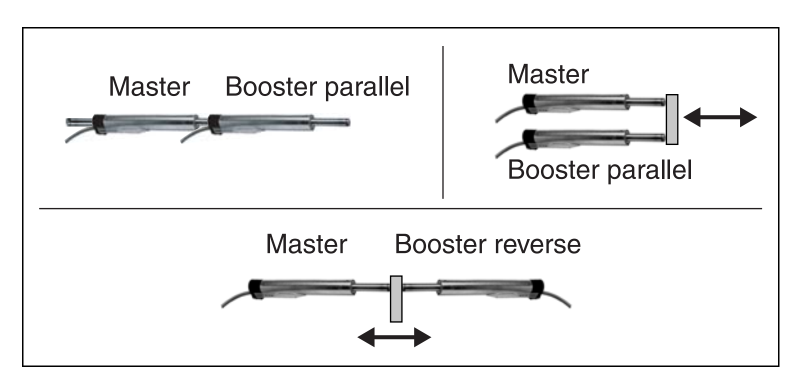


Abbildung 5-2: Booster-Betriebsarten

5.2 Master/Gantry operation

Ähnlich wie bei Master/Booster Betrieb arbeiten auch beim Master/Gantry Betrieb mehrere Motoren zusammen, wobei auch hier nur der Master Motor angesteuert wird. Im Unterschied zum Master/Booster betrieb werden aber hier alle Motoren positionsgeregelt, was eine an Konstruktions ermöglicht, wo die Motoren weit auseinander liegen und miteinander mechanisch nur schwach gekoppelt sind.

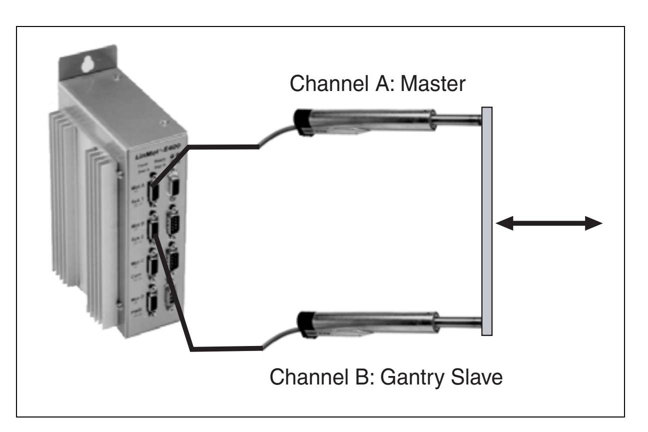


Abbildung 5-3: Gantry Betrieb

Funktionsprinzip

Die Sollbewegung wird nur für den Master Motor vorgegeben und auf die Gantry Slaves automatisch kopiert. Die Motoren sind unabhängig voneinander positionsgeregelt.

Parameter

Der Master Motor muss vor den Slaves definiert werden. Die Slaves müssen komplett konfiguriert werden ausser dem Run Mode. Ein Slave wird definiert mit der Auswahl “Gantry Slave parallel” unter “DriveX\Advanced\Master / Booster\”.

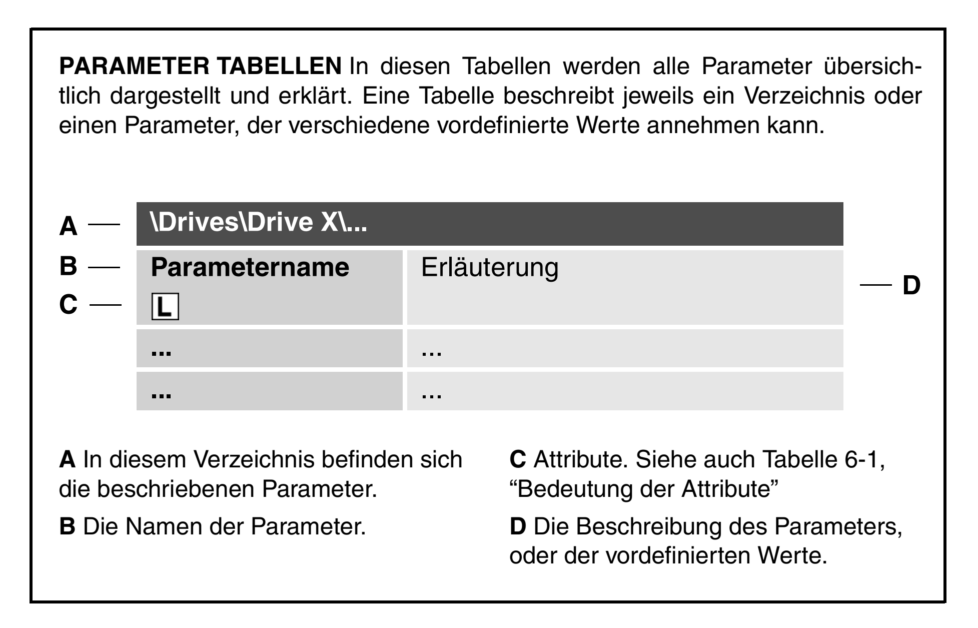


6. Parameter

6.1 Einführung

Alle Servo Controller der *LinMot*® Familie können mittels Parameter für die Anwendung konfiguriert werden. Die Konfiguration wird auf dem Controller im nichtflüchtigen EEPROM abgespeichert.

Die Parameter sind hierarchisch gruppiert (Baum-Struktur) und können mit dem **Parameter Inspector** einfach editiert werden. Alle Parameter werden in den folgenden Kapitel in tabellarischer Form aufgelistet und erklärt.



Alle **Parameter** und **Verzeichnisse** sind in diesem Kapitel der besseren Übersichtlichkeit halber **fett** geschrieben. Jeder Parameter kann zusätzliche Attribute besitzen. Die möglichen Attribute werden in der folgenden Tabelle erläutert.

Attribut	Bedeutung
L	Dieses Symbol kennzeichnet Live-Parameter. Diese können auch dann verändert werden, wenn die Motoren in Betrieb sind.
R	Dieses Symbol kennzeichnet Parameter welche schreibgeschützt sind und vom Anwender nicht verändert werden können.
	Dieses Symbol deutet darauf hin, dass der Parameter nur unter bestimmten Umständen sichtbar ist. In einer Fussnote am Ende der Tabelle wird angegeben wann der Parameter sichtbar ist.

Tabelle 6-1: Bedeutung der Attribute

Um eine bessere Übersicht zu gewährleisten, werden zusammengehörige Parameter in separaten Kapiteln beschrieben.

System-Parameter

Im Verzeichnis **System** wird das globale Systemverhalten definiert. Darunter fallen der Passwortschutz, die systembezogene Fehlerbehandlung, das Aufstartverhalten, die Systemzeit sowie die Versioneninformation.

Antriebs-Parameter

Im Verzeichnis **Drives** können bis zu vier Aktoren (Drive A, Drive B, Drive C, Drive D) unabhängig voneineander konfiguriert werden. Die Antriebs-Parameter definieren das Verhalten des betreffenden Aktors. Dieses wird bestimmt durch den Antriebstyp, die Initialisierung, das Generieren der Sollwerte, die Positionsüberwachung, die Regel-Parameter und die antriebsbezogene Fehlerbehandlung. Die Parameter für die Konfiguration der Linearmotoren, der Schrittmotoren, der Magnete und der externen Positionssensorik werden in separaten Kapiteln beschrieben.

MT-Parameter

Das Verzeichnis **Multi Trigger** dient zur Konfiguration der MT Servo Controller. Es ist nur sichtbar, wenn ein MT Servo Controller angeschlossen ist und wenn der Parameter **Command Interface** im Verzeichnis **System** den Wert **MT** hat.

PROFIBUS-Parameter

Im Verzeichnis **PROFIBUS** können Einstellungen für den PROFIBUS Servo Controller gemacht werden. Sie sind nur bei Controllern, welche über ein PROFIBUS-Interface verfügen, sichtbar (z.B. E430-DP).

Die folgende Tabelle zeigt in welchen Kapiteln die einzelnen Parameter beschrieben sind.

System-Parameter	Kapitel “System-Parameter” auf Seite 63
Antriebs-Parameter	Kapitel “Antriebs-Parameter” auf Seite 70
Linearmotor	Kapitel “Linearmotor-Parameter” auf Seite 72
Schrittmotor	Kapitel “Schrittmotor-Parameter” auf Seite 87
Magnet	Kapitel “Magnet-Parameter” auf Seite 96
Positionssensorik	Kapitel “Positionssensorik-Parameter” auf Seite 100
MT-Parameter	Kapitel “MT-Parameter” auf Seite 101
PROFIBUS-Parameter	Kapitel “PROFIBUS-Parameter” auf Seite 103



6.2 System-Parameter

System-Parameter definieren das globale Verhalten des Systems. Darunter fallen die Passwortbehandlung, die Fehler-Behandlung, das Aufstartverhalten, die Ein-/Ausgangs-Konfiguration, die Systemzeit sowie die Systeminformation.

Systeminformationen

Device

Die Parameter im Verzeichnis **Device** geben Auskunft über die Hardware des Systems.

\System\Info\Hardware\Device R	
Type	Gibt Auskunft über den Controller-Typ
Serial No High	Die oberen drei Ziffern der Seriennummer
Serial No Low	Die unteren drei Ziffern der Seriennummer
Serial Number	Die Seriennummer im Stringformat
Article Number	Die Artikelnummer im Stringformat

Memory

Die Parameter im Verzeichnis **Memory** geben Auskunft über den Speicher des Servo Controllers.

\System\Info\Hardware\Memory R	
Flash Type	Typ des eingesetzten FLASH-EPROM-Speichers
EEPROM Type	Typ des eingesetzten EEPROM-Speichers
RAM Type	Typ des eingesetzten RAM-Speichers

Software

Die Parameter im Verzeichnis **Software** beschreiben die auf dem Servo Controller installierte Software.

\System\Info\Software R	
Release	Softwarerelease
Monitor	Version des Monitors
Base	Version der Basissoftware
Application	Version der Applikationssoftware
Application 2	Version der 2. Applikationssoftware
Tree Type	Parameterbaum-Typ
Tree Version	Version des Parameterbaumes

ID Switch Position

Dieser Parameter zeigt die Stellung der ID-Schalter an.

\System\Info	
ID Switch Position [L]	Dieser Wert zeigt die Stellung der beiden hexadezimalen Drehcodierschalten auf dem Kontroller an (auf Frontseite beim DP-Kontroller oder Version 3 Kontroller, sonst auf der Unterseite). Die Schalter werden bei Bus-Interfaces zur Adresseinstellung verwendet.

Passwörter

Passwords

Im Verzeichnis **Passwords** kann das Passwort für den Servo Controller gesetzt werden.

\System\Passwords	
User	Der Parameter **User** enthält das aktuell gesetzte User-Passwort. Es kann nur an dieser Stelle verändert werden. Die maximale Länge für Passwörter ist 9 Zeichen. Bei der Auslieferung der Servo Controller ist kein Passwort eingestellt.

Fehler-Behandlung

In diesem Unterkapitel wird die systemseitige Fehlerbehandlung beschrieben. Bei der Fehlerbehandlung wird unterschieden zwischen Warnung und Fehler. In den meisten Fällen wird beim Auftreten eines Fehlers zuerst eine Warnung generiert. Reagiert das übergeordnete System nicht auf eine anstehende Warnung, geht der Servo Controller in den Fehlerzustand.

Warnungen

Eine Warnung äussert sich mit einem aktiven digitalen Ausgangssignal **WARNING OUT**. Dieses wird beim Auftreten einer Warnung aktiviert. Beim Ausgeben einer Warnmeldung ist das System noch voll einsatzfähig. Alle Motoren werden geregelt. Eine Warnmeldung gibt einem übergeordneten System die Möglichkeit, auf die Warnung zu reagieren und das System kontrolliert auszuschalten (z.B. in eine Notstop-Position zu fahren). Bei einer Warnung befindet sich der Servo Controller immer noch im Betriebszustand **RUN**.

Fehler

Ein Fehler äussert sich einerseits durch das Aktivieren des digitalen Ausgangssignal **ERROR OUT**. Dieses wird beim Auftreten eines Fehlers aktiviert. Andererseits führt ein Fehler zu einem unverzüglichen Abschalten aller vom Servo Controller gesteuerten Motoren. Der Controller ist dann im Fehlerzustand **ERROR**.

Bei fatalen Fehlern kann es vorkommen, dass ohne Warnung direkt in den Fehlerzustand gesprungen wird.



Error Mask[1]

Die selektierten Einträge bestimmen, welche Fehler die Basissoftware veranlassen, in den Fehlerzustand zu gehen. Bei einem Fehler werden die Motoren nicht mehr angesteuert.

\System\Error Handling\Error Mask R	
DCLV Power Too Low	Die Versorgungsspannung für den Leistungsteil ist zu tief.
DCLV Power Too High	Die Versorgungsspannung für den Leistungsteil ist zu hoch.
DCLV Signal Too Low	Die Versorgungsspannung für den Signalteil ist zu tief.
DCLV Signal Too High	Die Versorgungsspannung für den Signalteil ist zu hoch.
Electronic Fault	Der Kühlkörper des Controllers ist zu heiss (über 70°) oder ein Kurzschluss bei einer Motorphase ist detektiert worden.

Warn Mask[1]

In diesem Verzeichnis können alle internen Fehler selektiert werden, die zu einer Warnung (Warnsignal am digitalen Ausgang) führen sollen.

\System\Error Handling\Warn Mask	
DCLV Power Low	Die Versorgungsspannung für den Leistungsteil ist tief.
DCLV Power High	Die Versorgungsspannung für den Leistungsteil ist hoch.
DCLV Signal Low	Die Versorgungsspannung für den Signalteil ist tief.
DCLV Signal High	Die Versorgungsspannung für den Signalteil ist hoch.
Electronic Fault	Der Kühlkörper des Controllers ist heiss[1] (über 70°) oder ein Kurzschluss bei einer Motorphase ist detektiert worden.

1) Sobald die Temperatur des Kühlkörpers 70° überschreitet wird die Warnung **Electronic Fault** generiert. Nach 5 Sekunden wird dann der Fehler **Electronic Fault** generiert.

1) Die genauen Grenzwerte für die Überwachungsspannungen sind in der Tabelle 6-3 auf Seite 68 angegeben.

Msg Mask

Die Parameter im **Msg Mask** Verzeichnis bestimmen wann das Signal **Msg Output** aktiv sein soll (Parameter selektiert) oder inaktiv.

Logging Mask

Die Parameter im Logging Mask Verzeichnis bestimmen welche Fehler im Fehlerlog des Controllers abgespeichert werden sollen. Das Fehlerlog bleibt auch bei einem Spannungsausfall erhalten.

\System\Error Handling\Msg Mask [1] **\System\Error Handling\Logging Mask**	
DCLV Power Too Low	Die Versorgungsspannung für den Leistungsteil ist zu tief.
DCLV Power Too High	Die Versorgungsspannung für den Leistungsteil ist zu hoch.
DCLV Signal Too Low	Die Versorgungsspannung für den Signalteil ist zu tief.
DCLV Signal Too High	Die Versorgungsspannung für den Signalteil ist zu hoch.
Electronic Fault	Der Kühlkörper des Controllers ist über 70° heiss oder ein Kurzschluss bei einer Motorphase ist aufgetreten.
Drive Type Mismatch	Der angeschlossene Aktor-Typ ist nicht identisch mit dem mit dem eingestellten Aktor-Typ.
Curve Error	Ein Bewegungsprofil kann nicht gefunden werden oder passt nicht mit dem eingestellten Aktor-Typ zusammen.
Slider Missing	Bei einem Motor fehlt der Läufer, oder der Motor wurde nicht korrekt angeschlossen.
Init Failed	Der Initialisierungsvorgang konnte nicht erfolgreich abgeschlossen werden.
Drive Following Error	Ein Motor hat einen zu grossen Schleppfehler. Der Schleppfehler kann für jeden Motor separat eingestellt werden.
Drive Too Hot Calculated	Ein Motor ist kurzzeitig überlastet worden. Gründe dafür können sein: Motor wurde blokkiert oder der Motor wurde überlastet (zu grosse Lastmasse, zu schnelles Bewegungsprofil). Wird dieser Parameter deaktiviert, kann es sein, dass eine Überhitzung des Motors, welche durch eine kurzzeitige Überlast entsteht, nicht mehr detektiert wird. Der Motor kann Schaden nehmen.
Drive Too Hot Sensor	Ein Motor ist zu warm. Er ist überlastet und/oder zu schlecht gekühlt.

1) Nicht sichtbar wenn im Verzeichnis **\System\Command Interface** der Parameter **MT** angewählt wurde.

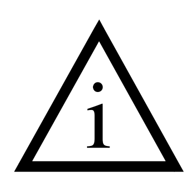


DCLV Monitoring

Die Warnungen und Fehler bei der Spannungsüberwachung werden anhand der in diesem Verzeichnis aufgeführten Werte generiert.

\System\Error Handling\DCLV Monitoring R	
Power Low Warn	Beim Unterschreiten dieser Spannung auf der Leistungsversorgung wird eine Warnung ausgelöst.
Power High Warn	Beim Überschreiten dieser Spannung auf der Leistungsversorgung wird eine Warnung ausgelöst.
Power Low Error	Beim Unterschreiten dieser Spannung auf der Leistungsversorgung wird ein Fehler ausgelöst.
Power High Error	Beim Überschreiten dieser Spannung auf der Leistungsversorgung wird ein Fehler ausgelöst.
Signal Low Warn	Beim Unterschreiten dieser Spannung auf der Signalversorgung wird eine Warnung ausgelöst.
Signal High Warn	Beim Überschreiten dieser Spannung auf der Signalversorgung wird eine Warnung ausgelöst.
Signal Low Error	Beim Unterschreiten dieser Spannung auf der Signalversorgung wird ein Fehler ausgelöst.
Signal High Error	Beim Überschreiten dieser Spannung auf der Signalversorgung wird ein Fehler ausgelöst.

Standardmässig wird eine Warnung erzeugt, wenn die Nominalspannung um 6% überschritten oder um 12% unterschritten wird. Ein Fehler, der zwangsweise zu einem Herunterfahren des Servo Controllers führt, wird bei einer Überschreitung der Nominalspannung von 12% oder einer Unterschreitung von 24% ausgelöst. Diese Schwellen können nicht vom Anwender eingestellt werden.



Wir empfehlen vor und während der ersten Inbetriebnahme die Spannungsversorgung zu überprüfen. Dabei sollte vor allem überprüft werden, ob das angeschlossene Netzteil genügend Strom und Spannung liefert.

		Nominal	**Default Warnschwelle [V]**	**Default Fehlerschwelle [V]**
Signalspeisung 24-48V	Min.	24V	21.12V	18.24V
	Max.	48V	50.88V	53.76V
Powerspeisung 24-48V	Min.	24V	21.12V	18.24V
	Max.	48V	50.88V	53.76V
Powerspeisung 48-72V	Min.	48V	42.24V	36.48V
	Max.	72V	76.32V	77.93V

Tabelle 6-2: Standardschwellwerte für die Versorgungspannung bei Version 2 Kontrollern

		Nominal	Default Warnungs-schwelle [V]	Default Fehler-schwelle [V]
Signalspeisung 24-48V	Min.	24V	21.12V	18.25V
	Max.	48V	50.86V	53.73V
Powerspeisung 48-72V	Min.	48V	42.24V	36.50V
	Max.	80V	90.23V	92.29V

Tabelle 6-3: Standardschwellwerte für die Versorgungspannung bei Version 3 Kontrollern

Aufstartverhalten

Startup Mode

Im **Startup Mode** Verzeichnis existiert ein Ein/Aus-Parameter, mit dem das Aufstartverhalten der Basissoftware bestimmt werden kann.

\System\Startup Mode	
Auto Start	Ist dieser Parameter gesetzt, springt der Controller nach dem Anlegen der Versorgungsspannung automatisch in den Zustand **INIT** und danach in den Zustand **RUN**, sofern kein Fehler und keine Warnung aufgetreten sind.
Init Together	Ist dieser Parameter gesetzt werden alle aktiven Motoren miteinander initialisieren, sonst beginnt Motor A mit initialisieren, danach B, C und D.

Ein-/Ausgangs-Konfiguration

IO Configuration

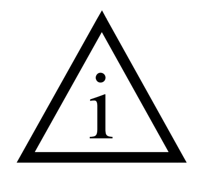
Mit den Parametern in diesem Verzeichnis wird definiert, welche Ein- oder Ausgangssignale der Standardschnittstelle aktiviert sind. Alle aktivierten Ein- und Ausgänge werden eingelesen oder beschrieben.

\System\IO Configuration	
Run Input	Diese Parameter bestimmen, ob die Signale von der Schnittstelle eingelesen werden.
Init Input	
Freeze Input	
Emerg Stop Input	
Analog / Trig Drive A	Diese Parameter bestimmen, ob die Triggersignale in den Zuständen **DRIVE INIT** und **RUN** eingelesen werden.
Analog / Trig Drive B	
Analog / Trig Drive C	
Analog / Trig Drive D	



\System\IO Configuration	
Error Output	Diese Parameter bestimmen, ob die entsprechenden Signale auf die Schnittstelle ausgegeben werden.
Warn Output	
Pos Error Output [1]	
Msg Output [1]	

1) Nicht sichtbar wenn im Verzeichnis **\System\Command Interface** der Parameter **MT** angewählt wurde



Im Normalfall sollten nur die Ein- bzw. Ausgänge selektiert werden, die von der übergeordneten Steuerung benötigt, resp. verarbeitet werden.

Befehls-Schnittstelle

Command Interface

In dem **Command Interface** Verzeichnis steht, von welcher Schnittstelle die Befehle für den Servo Controller vorgegeben werden. Es kann gleichzeitig nur eine Schnittstelle aktiviert werden.

\System\Command Interface	
AT	Die Befehle werden über das AT-Schnittstelle vorgegeben. Dies ist die Default-Einstellung.
MT	Die Befehle werden über die MT-Schnittstelle vorgegeben. Diese Schnittstelle darf nur angewählt werden, wenn ein MT Servo Controller angeschlossen ist. Ansonsten wird beim Aufstarten des Controllers eine Fehlermeldung ausgegeben!
ASCII RS232	Die Befehle werden über die RS232-Schnittstelle mit Hilfe des ASCII-Protokolls vorgegeben.
ASCII RS485	Die Befehle werden über die RS485-Schnittstelle mit Hilfe des ASCII-Protokolls vorgegeben.
Application	Die Befehle werden über eine spezielle Applikationssoftware vorgegeben. Diese Applikationssoftware kann für spezielle Anwendungen von der Sulzer Electronics AG programmiert werden.

Systemzeit

Time

In diesem Verzeichnis kann die Systemzeit, aufgeteilt in Betriebsstunden (Hours) und Sekunden (Seconds), abgefragt werden. Wird der Servo Controller über die Taste **Stop** im Control Panel angehalten, wird die Systemzeit bis zum nächsten Start angehalten.

\System\Time	
Hours	Betriebsstunden
Seconds	Anzahl Betriebssekunden (0…3600 s)

Regeneration Resistor

Die Schalthysterese des Abtaktwiderstandes wird hier eingestellt. Dieser Parameter existiert nur bei den Version 3 Kontrollern. Bei den MOT SUPPLY Schraubklemmen kann ein externer Abtaktwiderstand angeschlossen werden, welcher über den eingebauten MOSFET bei Bedarf zugeschaltet wird.

\System\Regeneration Resistor L	
Switch Off Voltage	Unterschreitet die Spannung der Leistungsspeisung diesen Wert wird der Abtaktwiderstand ausgeschaltet (MOSFET wird hochohmig).
Switch On Voltage	Überschreitet die Spannung der Leistungsspeisung diesen Wert wird der Abtaktwiderstand eingeschaltet (MOSFET durchgeschaltet).

6.3 Antriebs-Parameter

Die Aktor-Schnittstellen **Drive A** bis **Drive D** haben alle dieselben Parameter. Die nachfolgend beschriebenen Parameter sind dementsprechend für alle vier Aktor-Schnittstellen dieselben.

Die Antriebsparameter sind in Unterkapitel aufgeteilt, welche für die Definition von Antriebstyp, Initialisierung, Sollwertgenerierung, Positionsüberwachung, Steuerung und Fehlerbehandlung zuständig sind. Je nach eingestelltem Antriebstyp erscheinen unterschiedliche Unterverzeichnisse.

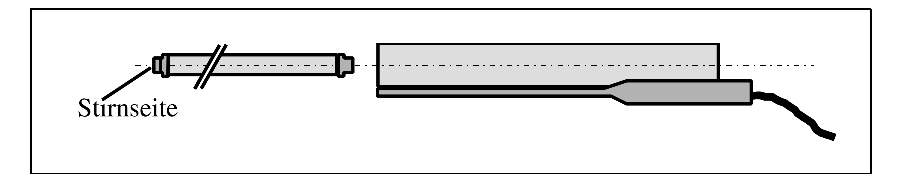
Type

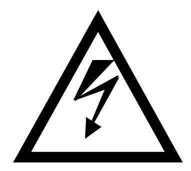
Dieser Parameter definiert den angeschlossenen Aktor-Typ. Es können die folgenden Aktoren gewählt werden:

\Drives\Drive X\Type	
No Drive	Dieser Typ muss gewählt werden, wenn kein Aktor angeschlossen ist, oder der Aktor ausgeschaltet werden soll.
***LinMot®* P0x-23**	Wenn ein *LinMot®* P0x-23 Antrieb angeschlossen ist, muss dieser Typ selektiert werden.
***LinMot®* P0x-23F**	Wenn ein *LinMot®* P0x-23F (fast) Antrieb angeschlossen ist, muss dieser Typ selektiert werden.
***LinMot®* P0x-37**	Wenn ein *LinMot®* P0x-37 Antrieb angeschlossen ist, muss dieser Typ selektiert werden.
***LinMot®* P0x-37 PL19**	Wenn ein *LinMot®* P0x-37 Antrieb in Kombination mit einem untermassigen Läufer angeschlossen ist, muss dieser Typ selektiert werden.
***LinMot®* P0x-37-HP**	Wenn ein *LinMot®* P0x-37-HP (high performance) Antrieb angeschlossen ist, muss dieser Typ selektiert werden.
***LinMot®* P0x-37-HP PL19**	Wenn ein *LinMot®* P0x-37-HP (high performance) Antrieb in Kombination mit einem untermassigen Läufer angeschlossen ist, muss dieser Typ selektiert werden.



\Drives\Drive X\Type	
***LinMot®* P0x-37F**	Wenn ein *LinMot®* P0x-37F (fast) Antrieb angeschlossen ist, muss dieser Typ selektiert werden.
***LinMot®* P0x-37F PL19**	Wenn ein *LinMot®* P0x-37F (fast) Antrieb in Kombination mit einem untermassigen Läufer angeschlossen ist, muss dieser Typ selektiert werden.
***LinMot®* P0x-48**	Wenn ein *LinMot®* P0x-48 Antrieb angeschlossen ist, muss dieser Typ selektiert werden.
***LinMot®* P0x-48 PL27**	Wenn ein *LinMot®* P0x-48 Antrieb in Kombination mit einem untermassigen Läufer angeschlossen ist, muss dieser Typ selektiert werden.
Stepper	Wenn ein Schrittmotor angeschlossen ist, muss dieser Typ gewählt werden. Der Schrittmotor muss zweiphasig sein.
Magnet	Um mit dem Servo Conteroller einen Elektromagnet anzusteuern, wird dieser Antriebstyp selektiert.
Sin/Cos Position Sensor	Wenn ein externer Positionssensor (sin/cos) an diesen Kanal angeschlossen werden soll muss dieser Typ selektiert werden. Siehe auch Kapitel “Externe Positionssensorik” auf Seite 54.





Es muss unbedingt darauf geachtet werden, dass die Konfiguration mit dem verwendeten Antriebstyp übereinstimmt. Eine falsche Konfiguration kann zur Zerstörung des angeschlossenen Antriebes (Linearmotor, Schrittmotor, Magnet) führen.

Die Aktoren **Schrittmotor** und **Magnet** besitzen im Normalfall keine eingebauten Temperatursensoren. Deshalb gibt es bei diesen keine Überwachungsfunktionen auf Überlast. Der Anwender ist selber verantwortlich, dass diese Antriebe im Betrieb nicht durch Überlastung (Übertemperatur) zerstört werden.

Die Parameter für die unterschiedlichen Antriebstypen werden in gesonderten Kapiteln erläutert:

Linearmotor	Kapitel “Linearmotor-Parameter” auf Seite 72
Schrittmotor	Kapitel “Schrittmotor-Parameter” auf Seite 87
Magnet	Kapitel “Magnet-Parameter” auf Seite 96
Positions Sensor	Kapitel “Positionssensorik-Parameter” auf Seite 100

6.4 Linearmotor-Parameter

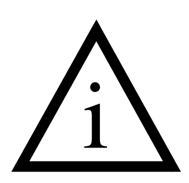
In diesem Kapitel werden alle Parameter erklärt, welche für die Parametrierung der *LinMot*® Linearmotoren benötigt werden. Damit diese Parameter im Parameterinspektor sichtbar werden, muss im Verzeichnis **\Drives\Drive X\Type** ein *LinMot*® Antrieb gewählt werden.

Master / Booster Betrieb

Master / Booster

Mit diesem Parameter wird definiert, ob der angeschlossene Motor im Master- oder im Booster-Modus betrieben wird. Ein Motor, der im Booster-Modus betrieben wird, übernimmt die meisten Einstellungen seines Masters. Dabei wird immer der vorherige Motor als Master angesehen. Wenn also **Motor B** als Booster konfiguriert wird, übernimmt er Einstellungen von **Motor A**. Eine ausführlichere Erklärung findet sich im Kapitel “Master/Slave-Betrieb” auf Seite 59.

\Drives\Drive X\Advanced\Master / Booster	
Master	Der angeschlossene Aktor ist ein Master.
Booster parallel	Der angeschlossene Aktor ist ein Booster, der parallel zum Master läuft.
Booster reverse	Der angeschlossene Aktor ist ein Booster, der entgegengesetzt zum Master läuft.
Gantry Slave parallel	Der angeschlossene Aktor ist ein Gantry Slave.



Es ist möglich den Master/Booster-Betrieb mit einer externen Positionssensorik zu kombinieren. Es können dabei maximal ein Positionssensor, ein Master und zwei Booster an einen *LinMot*® Servo Controller angeschlossen werden. Bei dieser Konfiguration muss am ersten Motorkanal die externe Positionssensorik, am zweiten Kanal der Master und an den weiteren Kanälen die Booster angeschlossen werden.



Positionssensorik

Position Sensor

Mit diesem Parameter bestimmt der Anwender von wo der Regler seine Positionsinformation bezieht. Wenn eine externe Positionssensorik benutzt wird, kann zusätzlich noch die Auflösung eingestellt werden.

\Drives\Drive X\Advanced\Position Sensor	
Internal Sensor	Bei dieser Einstellung wird die Istposition mit Hilfe der im *LinMot*® Motor eingebauten Positionssensorik bestimmt. Dies ist die Defaulteinstellung.
External 20µm **External 10µm** **External 5µm** **External 2.5µm** **External 1.25µm**	Bei dieser Einstellung bezieht der Motor seine Istposition von der am darüberliegenden Kanal angeschlossenen Positionssensorik. Ein Motor der z.B. am Kanal B angeschlossen ist, bezieht daher seine Istposition von der Sensorik welche am Kanal A angeschlossen ist.
AB Enc1 1X **AB Enc1 2X** **AB Enc1 4X** **AB Enc2 1X** **AB Enc2 2X** **AB Enc2 1X**	Mit dieser Einstellung bezieht der Motor seine Istpost vom Positionssensor, welcher am entsprechenden Link des Masterencoder Moduls hängt. Der Sensortyp ist inkrementell AB und die Dekokierungsmodi können 1X (nur steigende Flanke von A), 2X (alle steigenden Flanken von A and B) or 4X (alle Flanken von A and B).

Im Kapitel “Positionssensorik-Parameter” auf Seite 100 wird erklärt wie die eingestellte Auflösung mit dem Fahrbereich zusammenhängt.

Initialisierung

Initialization

Die in den Antrieben der *LinMot*® P Serie eingesetzte Positionssensorik erlaubt eine relative Bestimmung der Position. Bei einem Aufstartvorgang muss daher die Referenzposition einmalig initialisiert werden. Dies geschieht mittels einer sog. Referenzfahrt.

Die Initialisierungsart eines Motors wird durch den Parameter **Init Mode**, den Parameter **Init Switches** und die Parameter im Verzeichnis **Init Config** definiert.

Init Mode

Der Initialisierungs-Modus legt fest, auf welche Art der Positionsnullpunkt des Läufers definiert wird. Folgende Modi stehen zur Auswahl:

\Drives\Drive X\Initialization\Init Mode	
Actual Position	Die gerade aktuelle Position wird als Nullpunkt definiert.
Auto Move Out	Der Läufer bewegt sich bis an einen mechanischen Anschlag nach aussen. Diese Position wird als Nullpunkt gesetzt.
Auto Move In	Der Läufer bewegt sich bis an einen mechanischen Anschlag nach innen. Diese Position wird als Nullpunkt gesetzt.
Trig Move Out	Der Läufer bewegt sich nach aussen, bis durch die positive Flanke des Triggersignals der Nullpunkt festgelegt wird. Ist zu Beginn der Initialisierung das Triggersignal bereits gesetzt, bewegt sich der Läufer solange nach innen, bis das Triggersignal abfällt. Dann erst bewegt sich der Läufer wieder nach aussen, damit der Nullpunkt mit der positiven Triggerflanke festgelegt werden kann.
Trig Move In	Der Läufer bewegt sich nach innen, bis durch die positive Flanke des Triggersignals der Nullpunkt festgelegt wird. Ist zu Beginn der Initialisierung das Triggersignal bereits gesetzt, bewegt sich der Läufer solange nach aussen, bis das Triggersignal abfällt. Dann erst bewegt sich der Läufer wieder nach innen, damit der Nullpunkt mit der positiven Triggerflanke festgelegt werden kann.

Init Switches

\Drives\Drive X\Initialization\Init Switches	
Init Once	Falls dieser Parameter aktiviert ist, wird der Motor nur das erste Mal im Zustand **INIT** initialisiert. Wenn dieser Parameter nicht aktiviert ist, werden die Motoren jedes Mal initialisiert, sobald ein aktives **INIT**-Signal anliegt.



Init Config

Im Verzeichnis **Init Config** befinden sich folgende Initialisierungs-Parameter.

\Drives\Drive X\Initialization\Init Config	
Init Speed	Definiert die Geschwindigkeit, mit welcher sich der Läufer bei der Initialisierung bewegt.
Maximal Init Current	Dieser Parameter gibt an, ab welchem Strom der Linearmotor beim Initialisieren einen Anschlag detektieren soll. Es kann somit festgelegt werden, wie fest der Linearantrieb beim Initialisieren gegen den Anschlag drücken soll. Der Strom wird prozentual zum Maximalstrom vorgegeben.
Home Position	Nachdem beim Initialisieren der mit dem Parameter **Maximal Init Current** vorgegebene Strom erreicht wurde, gilt der Motor als initialisiert. Dieser Position wird nun der Wert **Home Position** zugewiesen.
Check Init Position	Nachdem die **Home Position** definiert ist, wird automatisch auf die unter **Check Init Position** angegebene Position und wieder zurück zur **Home Position** gefahren. Kann der Läufer die gewünschte Position nicht erreichen (gewünschter Hubbereich nicht frei), wird eine Fehlermeldung ausgegeben.
Initial Position	Am Ende der Initialisierung wird die **Initial Position** angefahren. Hat der Läufer diese Position erreicht, ist die Initialisierung abgeschlossen.

Im Beispiel auf der nächsten Seite wird die Bedeutung dieser Parameter nochmals verdeutlicht.

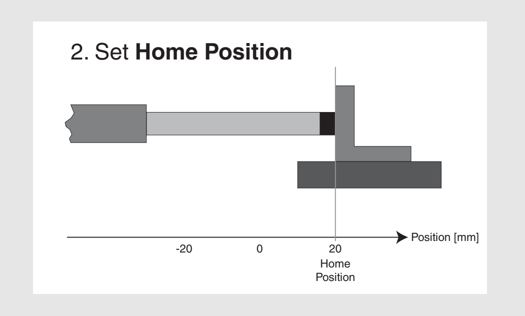
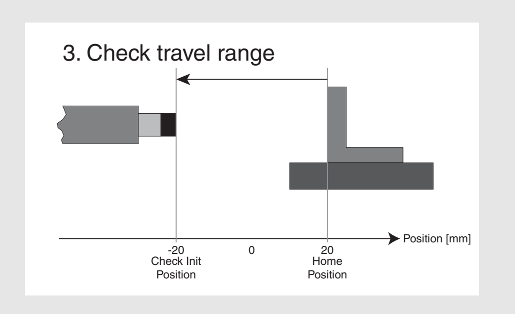
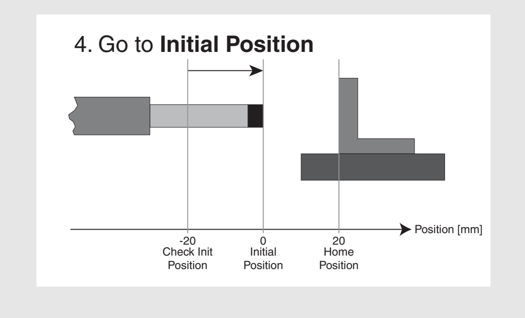
Das folgende Beispiel zeigt, wie eine Initialisierung abläuft. Die Initialisierungsparameter wurden dabei wie folgt gesetzt.

Init Mode: Auto Move Out

Home Position: 20.0 mm

Check Init Position: -20.0 mm

Initial Position: 0.0 mm

Aktion	Beschreibung
1. Search **Home Position** Position [mm]	Der Motor sucht die **Home Position**. Da als Initialisierungsmodus **Auto Move Out** gewählt ist, wird sich der Läufer aus dem Motor heraus bewegen. Die Suche wird abgeschlossen sobald der Strom den Wert **Maximal Init Current** erreicht. Damit kann eingestellt werden, wie fest der Motor gegen den Anschlag drükken soll.
	Nachdem der Motor die **Home Position** gefunden hat, wird dieser Position der Wert **Home Position**, in unserem Beispiel also 20 mm, zugewiesen. Damit ist die Positionsachse definiert.
	Nachdem die **Home Position** definiert wurde, wird die **Check Init Position** angefahren. Wenn bei dieser Überprüfungsfahrt ein Fehler auftritt wird die Initialisierung abgebrochen. Wenn keine Überprüfung des Fahrbereichs gewünscht wird, muss die **Check Init Position** gleich der **Initial Position** gesetzt werden.
	Nachdem der Fahrbereich überprüft worden ist, wird die **Initial Position** angefahren. Nachdem diese erreicht worden ist, wird der Initialisierungsvorgang abgeschlossen und der Motor ist betriebsbereit.



Generierung der Sollwerte

In diesem Verzeichnis wird die Art der Sollwertvorgabe eingestellt.

Run Mode

Der Modus der Sollwertvorgabe wird bei AT- und MT Controllern mit dem **Run Mode**-Parameter definiert. Dabei können die folgenden Modi unterschieden werden:

\Drives\Drive X\Set Value Generation\Run Mode	
Serial	Der Sollwert wird mit einem Protokoll über die serielle Schnittstelle vorgegeben.
Analog	Der Sollwert wird über den dem Motor entsprechenden analogen Eingang vorgegeben. Im Verzeichnis **Set Value Generation** können die Randsollwerte für die Eingangspegel 0V und 10V festgelegt werden. Der analoge Eingangsbereich zwischen 0V und 10V wird linear auf den Sollwertbereich abgebildet, der durch diese Randsollwerte (Maximal-/Minimalwert) definiert ist.
Continuous Curve	Eine auf dem Servo Controller abgespeicherte Sollwertkurve wird zyklisch abgefahren. Die Kurvennummer wird unter **Curve Number** im Verzeichnis **Set Value Configuration** gewählt.
Trigger Curve	Auf die steigende Flanke des Triggersignals wird eine erste, auf die fallende Flanke eine zweite auf dem Servo Controller gespeicherte Kurve abgefahren. Die Kurvennummern werden im Verzeichnis **Set Value Generation** unter **Rise Curve Number** und **Fall Curve Number** festgelegt. Folgt die fallende Flanke des Triggersignals noch bevor die erste Kurve zu Ende ist, wird die zweite Kurve unmittelbar anschliessend abgefahren.
Two Point	Ist das dem Antrieb entsprechende Triggersignal aktiv, wird der Wert 'angefahren', welcher mit dem Parameter **'1' Position** im Verzeichnis **Set Value Generation** definiert ist. Ist das Triggersignal nicht aktiv, wird der Wert 'angefahren', welcher mit dem Parameter **'0' Position** definiert werden kann.

Set Value Configuration

Im Verzeichnis **Set Value Configuration** sind alle Konfigurations-Parameter zusammengefasst, die im Umgang mit der Sollwertvorgabe benötigt werden.

\Drives\Drive X\Set Value Generation\Set Value Configuration	
Minimal Position L	Bestimmt die minimale Ist-Position (Untergrenze vom Positionierbereich).
Maximal Position L	Bestimmt die maximale Ist-Position (Obergrenze vom Positionierbereich).
0V Position L 👁[1]	Dieser Parameter definiert im Modus **Analog** die Position, welche bei einer Eingangsspannung von 0 V angefahren werden soll.
10V Position L 👁[1]	Dieser Parameter definiert im Modus **Analog** die Position, welche bei einer Eingangsspannung von 10 V angefahren werden soll.
'0' Position L 👁[2]	Definiert die Position, welche im Modus **Two Point** angefahren wird, wenn der Trigger auf einer logischen Null liegt.
'1' Position L 👁[2]	Definiert die Position, welche im Modus **Two Point** angefahren wird, wenn der Trigger auf einer logischen Eins liegt.
Curve Number L 👁[3]	Im Modus **Continuous Curve** wird diejenige auf dem Servo Controller gespeicherte Kurve zyklisch 'abgefahren', welche die in **Curve Number** aufgeführte Nummer aufweist.
Rise Curve Number L 👁[4]	Im Modus **Trigger Curve** wird die Kurve mit dieser Nummer auf die steigende Flanke des Triggersignals abgefahren.
Fall Curve Number L 👁[4]	Im Modus **Trigger Curve** wird die Kurve mit dieser Nummer auf die fallende Flanke des Triggersignals abgefahren.
Curve Pos Offset L	Mit diesem Parameter kann der Positionsoffset der Kurve bestimmt werden.
Curve Amplitude L	Mit diesem Parameter kann die Amplitude der Kurve bestimmt werden. Der Wertebereich geht von 0 bis 100%.
Curve Speed L	Mit diesem Parameter kann die Geschwindigkeit der Kurve bestimmt werden. Der Wertebereich geht von 0 bis 100%.

1) Nur sichtbar im Modus **Analog**
2) Nur sichtbar im Modus **Two Point**
3) Nur sichtbar im Modus **Continuous Curve**
4) Nur sichtbar im Modus **Trigger Curve**



Die folgende Grafik verdeutlicht die Abbildung der analogen Eingangsspannung auf den Sollwert (Position/Strom) im **Analog**-Modus.

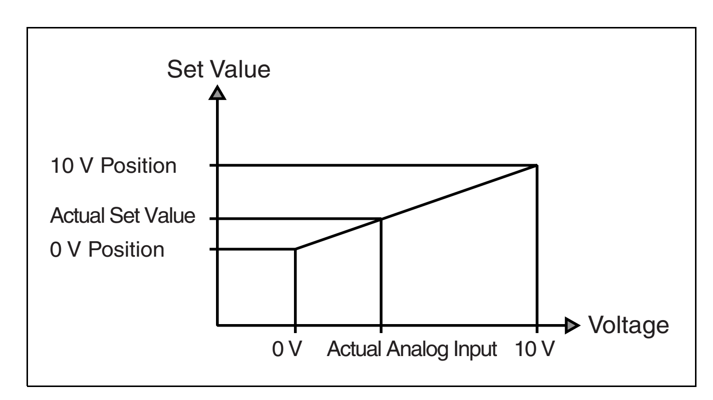


Abbildung 6-1: Sollposition bei analoger Sollwertvorgabe

Der errechnete Sollwert ist der zwischen dem Parameter **0V Position** und **10V Position** linear interpolierte Wert.

Filter Parameter

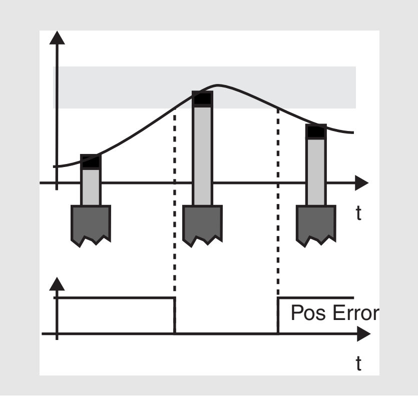
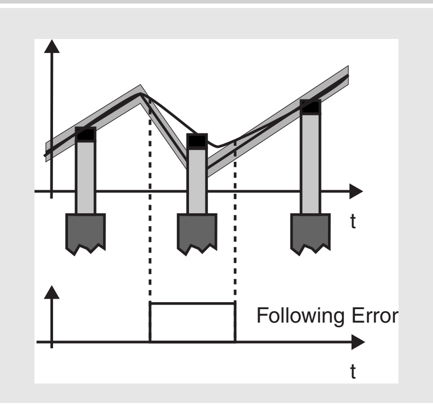
In diesem Verzeichnis werden alle Parameter aufgeführt, die für die Sollwert-Filterung notwendig sind.

\Drives\Drive X\Set Value Generation\Filter Parameter	
Max Speed L	Dieser Wert setzt die obere Grenze für die Geschwindigkeit. Er wird **nicht berücksichtigt**, wenn eine **Kurve** abgefahren wird! Beim Abfahren einer Kurve wird die Geschwindikeit und Beschleunigung aus der Kurve entnommen.
Max Accelera-tion L	Dieser Wert setzt eine obere Grenze für die Beschleunigung. Er wird **nicht berücksichtigt**, wenn eine **Kurve** abgefahren wird! Beim Abfahren einer Kurve wird die Geschwindikeit und Beschleunigung aus der Kurve entnommen.

Positionsüberwachung

Position Monitoring

Die Basissoftware unterstützt zwei Arten von Positionsüberwachung.

Positionsband-Überwachung	
Für jeden Motor existiert ein Positionsband. Befindet sich ein eingeschalteter Motor ausserhalb seines Bandes, wird der digitale Ausgang **POSITION ERROR OUT** aktiv.	
Schleppfehler-Überwachung	
Die Differenz zwischen Soll- und Ist-Position darf sich nur in gewissen Schranken bewegen. Wird diese Differenz zu gross, wird, falls selektiert, eine Warnung bzw. ein Error erzeugt. Gründe dafür können sein: zu schnelle Sollwertkurven, Sprünge in den Sollwertkurven, zu grosse Lastmasse, Motor verklemmt…	

Folgende Parameter dienen zur Spezifikation der Schranken:

\Drives\Drive X\Position Monitoring L	
Pos Range Min **Pos Range Max**	Definiert die obere resp. untere Schranke der Positionsbandüberwachung. Liegt die Ist-Position des Antriebs unterhalb dieses Wertes, wird der digitale Ausgang **POSITION ERROR OUT** aktiviert.
In Position- **In Position+**	Diese zwei Parameter definieren wie nahe sich der Motor an der gewünschten Zielposition befinden muss, damit die Zielposition als erreicht gilt. Nur aktiv in den Interface-Modi MT, ASCII und PROFIBUS! Wird einem Motor bei einem Zustandswechsel eine neue Sollposition vorgegeben (Relativ- oder Absolutbewegung) oder eine eine Kurve gestartet, geht das **In Position**-Signal des betreffenden Motors so lange auf logisch 0, bis sich der Läufer des Motors in einem Bereich um den Sollwert befindet.
Following Error- **Following Error+**	Definiert den maximal zulässigen Schleppfehler. Falls die Differenz zwischen Soll- und Ist-Position grösser resp. kleiner ist als der eingestellte Wert, wird eine Warnung bzw. ein Fehler generiert.



Regel- und Steuer-Parameter

Control Parameter

In diesem Verzeichnis sind jene Parameter aufgeführt, welche bei der Regelung und Steuerung der Antriebe benötigt werden. Siehe auch Kapitel “Tips und Tricks zum Regler” auf Seite 105.

\Drives\Drive X\Control Parameters L	
Maximal Current	Bestimmt den maximalen Strom, der vom Regler eingestellt werden kann. Dieser Parameter bestimmt durch den maximalen Strom auch die maximale Kraft.
Current Offset	Bestimmt den Stromoffset. Mit Hilfe dieses Parameters kann eine statisch anliegende Kraft am Motor kompensiert werden.
P	Bestimmt, wie die Differenz von der Soll- zur Istposition auf den Sollstrom abgebildet werden soll. Die Einstellung ‘1A/mm’ bewirkt, dass bei einer Positionsabweichung von 1mm ein Sollstrom von 1 Ampère resultiert.
D	Bestimmt, wie die Differenz der Soll- zur Istgeschwindigkeit auf den Sollstrom abgebildet werden soll. Die Einstellung ‘4A*s/mm’ bewirkt, dass bei einer Geschwindigkeitsdifferenz von 1mm/s ein Sollstrom von 4 Ampère resultiert.
I	Bestimmt, wie das Zeitintegral der Positionsabweichung auf den Sollstrom abgebildet werden soll. Die Einstellung ‘100A/(mm*s)’ hat zur Folge, dass bei einer Positionsdifferenz von 0.1mm die über eine Zeit von 0.1s anliegt ein Sollstrom von 1 Ampère eingestellt wird.



\Drives\Drive X\Control Parameters L	
FF Friction	Bestimmt, welcher konstante Strom bei einer Bewegung des Läufers vorgesteuert werden soll. Das Vorzeichen des vorgesteuerten Stromes ist abhängig von der Bewegungsrichtung des Motors. Dieser Parameter wird zur Kompensation einer evtl. vorhandenen Reibung eingesetzt.
FF Acceleration	Bestimmt, welcher Strom vorgesteuert werden muss, um die gewünschte Beschleunigung zu erreichen. Ein Wert von ‘100 mA/(m/s^2)’ führt dazu, dass bei einer Beschleunigungsdifferenz von 1m/s^2 ein Sollstrom von 100mA vorgesteuert wird.
FF Deceleration	Bestimmt, welcher Strom vorgesteuert werden muss, um die gewünschte Verlangsamung zu erreichen. Ein Wert von ‘100 mA/(m/s^2)’ führt dazu, dass bei einer Verlangsamungsdifferenz von 1 m/s^2 ein Sollstrom von -100 mA vorgesteuert wird.
Noise Dead Band	Dieser Parameter bestimmt die Breite des Geräuschfiltertotbandes. Diese Funktion dient zur Reduktion der Motorgeräusche im Stillstand. Weil dieses Filter die Poisitioniergenauigkeit reduziert, sollte es nur eingeschaltet werden, wenn die Geräusche störend sind und, falls eingeschaltet, sollte der Wert auf das Minimum eingestellt werden, wo die Geräusche verschwinden. Das Geräuschfilter wird aktiv, wenn die Sollposition sich nicht mehr verändert. Wenn der Integralanteil des Positionsreglers auf Null gesetzt ist, wird dieses Filter den Motorstrom fixieren, bis die Positionsabweichung zum Mittelwert der acht Ist-Positionen bevor das Filter aktiv wurde grösser als der eingegebene Wert wird. Ist der Integralanteil des Positionsreglers eingeschaltet, fixiert dieses Filter den Motorstrom, bis die Positionsabweichung zur Sollposition grösser als der eingestellt Wert wird.

Regel- und Steuermodi

Control Switches

Dieser Parameter ist nur bei Servo Controllern der E100-Serie sichtbar und bestimmt, ob mit 2 oder mit 3 Ampère Maximalstrom gearbeitet werden soll. Mit dem Parameter **Maximal Current** im Verzeichnis Control Parameter kann der Maximalstrom dann noch weiter eingegrenzt werden.

\Drives\Drive X\Control Switches	
Current ( ) 2A (x) 3A [1]	Wenn dieser Parameter selektiert ist, wird der Maximalstrom auf 3 Ampère, ansonsten auf 2 Ampère, begrenzt.

1) Nur sichtbar bei den Servo Controllern der Serie E100

Kommutierung

Commutation

In diesem Verzeichnis sind jene Parameter aufgeführt, welche die Kommutierung der angeschlossenen Antriebe bestimmen. Die Kommutierung bestimmt, wie die zwei Phasen des angeschlossenen Antriebes bestromt werden. Die Kommutierung hat beim Linearmotor einen Einfluss auf die folgenden Betriebsgrössen:

- Welligkeit der Kraft
- Wärmeverluste im Motor
- Dynamik des Motors

Es können die folgenden Kommutierungsarten gewählt werden:

\Drives\Drive X\Advanced\Commutation	
Sine (Default)	Welligkeit der Kraft: Klein Wärmeverluste im Motor: Klein Dynamik des Motors: Mittel In den **meisten Fällen** empfiehlt es sich, diese Standardeinstellung als Kommutierung zu wählen!
Trapezoid	Welligkeit der Kraft: Gross Wärmeverluste im Motor: Gross Dynamik des Motors: Mittel bis gross Diese Kommutierung sollte gewählt werden wenn eine grössere Dynamik als mit der Sinus-Kommutierung erreicht werden soll.
Block	Welligkeit der Kraft: Sehr gross Wärmeverluste im Motor: Sehr gross!!! Dynamik des Motors: Gross Diese Kommutierung darf wegen der sehr grossen Wärmeverluste nur gewählt werden, wenn höchste Dynamik bei einer kleinen Einschaltdauer gefordert ist. Die Einschaltdauer bezeichnet das Verhältnis zwischen der Bewegungszeit und der Stillstandszeit des Läufers.

Die Parameter für den Maximalstrom befinden sich im Verzeichnis **\Drives\Drive X\Control Parameters**.

Fehlerbehandlung

In diesem Unterkapitel wird die antriebsseitige Fehlerbehandlung beschrieben. Allgemeine Informationen zu Fehlern und Warnungen können dem Kapitel “Fehler-Behandlung” auf Seite 64 entnommen werden.

Error Mask
Warn Mask

In den unten aufgeführten Verzeichnissen kann für die Antriebe angegeben werden, bei welchen internen Fehlern in den Fehlerzustand gegangen wird (**Error Mask**), bei welchen Fehlern eine Warnung generiert wird (**Warn Mask**), wie die Temperaturüberwachung der Antriebe konfiguriert ist (**Temperature**) und was im Not-Stop-Fall geschehen soll (**Emergency Stop**)..

\Drives\Drive X\Error Handling\Error Mask	
Drive Type Mismatch R	Der angeschlossene Motorentyp stimmt nicht mit dem gewählten Motorentyp überein, oder der angeschlossene Motor ist defekt.
Curve Error R	Eine Sollwertkurve kann nicht gefunden werden.
Slider Missing R	Der Läufer fehlt, oder der Motor wurde nicht korrekt angeschlossen.
Init Failed R	Bei der Initialisierung des Motors ist ein Fehler aufgetreten.
Drive Following Error	Ein Motor hat einen zu grossen Schleppfehler.
Drive Too Hot Calculated	Ein Motor ist kurzzeitig überlastet worden. Gründe dafür können sein: Motor wurde blokkiert oder der Motor wurde überlastet (zu grosse Lastmasse, zu schnelles Bewegungsprofil). Wird dieser Parameter deaktiviert, kann es sein, dass eine Überhitzung des Motors, welche durch eine kurzzeitige Überlast entsteht, nicht mehr detektiert wird. Der Motor kann Schaden nehmen.
Drive Too Hot Sensor R	Es wurde eine zu starke Erwärmung des Motors festgestellt. Der Motor wurde überlastet und/oder zu schlecht gekühlt.



Die meisten Parameter im Verzeichnis **Warn Mask** können im Gegensatz zu denen im Verzeichnis **Error Mask** editiert werden. Einzig die Warnung **Drive Init Not Done** kann nicht gelöscht werden. Diese Warnung liegt an bis ein Motor initialisiert wurde und verhindert, dass der Servo Controller in den Zustand **RUN** gehen kann solange ein Motor noch nicht initialisiert wurde.

\Drives\Drive X\Error Handling\Warn Mask	
Slider Missing	Der Läufer fehlt, oder der Motor wurde nicht korrekt angeschlossen.
Drive Init Not Done R	Ein Motor hat nicht oder nicht korrekt initialisiert.
Pos Range Indicator	Der Läufer liegt ausserhalb des durch **Pos Range Min/Max** definierten Positionsbereichs. Wenn diese Warnung aktiviert wird muss darauf geachtet werden, dass die **Pos Range** Parameter so eingestellt werden, dass nach der Initialisierung das Signal nicht anliegt. Ansonsten kann nur schwer festgestellt werden ob alle Motoren korrekt initialisiert wurden. Der Ausgang wird bei einem AT Servo Controller nur aktiviert wenn der Parameter **Pos Range** im Verzeichnis **\System\IO Configuration** aktiviert ist. Bei einem MT Controller im MT-Modus (Parameter **MT** im Verzeichnis **\System\Command Interface** aktiviert) wird das Signal nur ausgegeben wenn der Parameter **Pos Range** im Verzeichnis **\Multi Trigger\Output Configuration\Output 3** aktiviert ist.
Drive Following Error	Ein Motor hat einen zu grossen Schleppfehler.
Drive Too Hot Calculated	Ein Motor ist kurzzeitig überlastet worden. Gründe dafür können sein: Motor wurde blokkiert oder der Motor wurde überlastet (zu grosse Lastmasse, zu schnelles Bewegungsprofil). Wird dieser Parameter deaktiviert, kann es sein, dass eine Überhitzung des Motors, welche durch eine kurzzeitige Überlast entsteht, nicht mehr detektiert wird. Der Motor kann Schaden nehmen.
Drive Hot Sensor	Es wurde eine zu starke Erwärmung des Motors festgestellt. Der Motor wurde überlastet und/oder zu schlecht gekühlt.
In Position	Ein Motor hat seine Sollposition erreicht. Der Ausgang **Msg Output** wird nur aktiviert wenn auch der Parameter **Msg Output** im Verzeichnis **\System\IO Configuration** aktiviert ist und der Parameter **In Position** im Verzeichnis **\System\Error Handling\Msg Mask** gesetzt ist.

Emergency Stop Mode

Dieser Parameter bestimmt, wie sich der Motor verhalten soll, nachdem der Eingang **STOP** aktiviert wurde.

\Drives\Drive X\Error Handling\Emergency Stop\Emergency Stop Mode	
Off	Der Motor wird nicht mehr angesteuert. Die Position wird weiterhin eingelesen. In diesem Modus werden die Wicklungen nicht mehr bestromt.
Freeze	Der Motor wird abgebremst[1] und in dieser Position gehalten.
Goto Position	Der Motor fährt[1] in die **Emergency Stop Position** und wird auf diese Position geregelt.

1) Die Geschwindigkeit und Beschleunigung dieser Bewegungen kann mit den Parametern **Max Speed** und **Max Acceleration** im Verzeichnis **Emergency Configuration** eingestellt werden

Emergency Configuration

Die Parameter in diesem Verzeichnis dienen zur Parametrierung des Fahrverhaltens beim Aktivieren des Eingangs STOP.

\Drives\Drive X\Error Handling\Emergency Stop\Emergency Configuration [1]	
Stop Position L [2]	Ist der Notaus-Modus **Drive Goto Position** selektiert, wird beim Aktivieren des Eingangs **STOP** unverzüglich die unter **Emergency Stop Position** definierte Position angefahren. Der Motor bleibt in Betrieb.
Max Speed L	Bestimmt, wie schnell sich der Antrieb in die **Stop Position** bewegt.
Max Acceleration L	Bestimmt, mit welcher Beschleunigung der Antrieb beim Aktivieren des Eingangs **STOP** abgebremst wird.

1) Nur sichtbar, wenn der Parameter **Emergency Stop Mode** auf **Goto Position** oder **Freeze** eingestellt ist.
2) Nur sichtbar wenn der Parameter **Emergency Stop Mode** auf **Goto Position** eingestellt ist.



6.5 Schrittmotor-Parameter

In diesem Kapitel werden alle Parameter erklärt, welche für die Parametrierung von Schrittmotoren benötigt werden. Damit diese Parameter im **Parameter Inspektor** sichtbar werden, muss im Verzeichnis **\Drives\Drive X\Type** der Parameter **Stepper** gewählt werden.

Master/Booster-Betrieb

Master/Booster

Dieser Parameter muss im Schrittmotormodus immer auf den Wert **Master** eingestellt werden. **Master/Booster**-Konfigurationen werden im Schrittmotormodus nicht unterstützt.

\Drives\Drive X\Advanced\Master / Booster	
Master	Muss angewählt werden, wenn ein Schrittmotor angesteuert wird.

Initialisierung

Initialization

Normale Schrittmotoren erlauben nur eine relative Bestimmung der Position. Bei einem Aufstartvorgang muss daher die Referenzposition einmalig initialisiert werden. Dies geschieht mittels einer sog. Referenzfahrt.

Die Initialisierungsart eines Motors wird durch den Parameter **Init Mode**, den Parameter **Init Switches** und die Parameter im Verzeichnis **Init Config** definiert.

Init Mode

Der Initialisierungs-Modus legt fest, auf welche Art der Positionsnullpunkt des Schrittmotors definiert wird. Folgende Modi stehen zur Auswahl:

\Drives\Drive X\Initialization\Init Mode	
Actual Position	Die gerade aktuelle Position wird als Nullpunkt definiert.
Trig Turn Left	Der Schrittmotor dreht sich solange im Gegenuhrzeigersinn, bis durch die positive Flanke des Triggersignals der Nullpunkt festgelegt wird. Ist zu Beginn der Initialisierung das Triggersignal bereits aktiv, bewegt sich der Rotor solange in die entgegengestzte Richtung, bis das Triggersignal abfällt. Dann erst bewegt sich der Rotor wieder links herum, damit der Nullpunkt mit der positiven Triggerflanke festgelegt werden kann.
Trig Turn Right	Der Schrittmotor dreht sich solange im Uhrzeigersinn, bis durch die positive Flanke des Triggersignals der Nullpunkt festgelegt wird. Ist zu Beginn der Initialisierung das Triggersignal bereits aktiv, bewegt sich der Rotor solange in die entgegengestzte Richtung, bis das Triggersignal abfällt. Dann erst bewegt sich der Rotor wieder rechts herum, damit der Nullpunkt mit der positiven Triggerflanke festgelegt werden kann.

Init Switches

\Drives\Drive X\Initialization\Init Switches	
Init Once	Falls dieser Parameter aktiviert ist, wird der Motor nur das erste Mal im Zustand **DRIVE INIT** initialisiert. Wenn dieser Parameter nicht aktiviert ist, werden die Motoren jedes Mal initialisiert, wenn vom **DISABLE**- in den **DRIVE INIT**-Zustand gewechselt wird.

Init Config

Im Verzeichnis **Init Config** befinden sich folgende Initialisierungs-Parameter.

\Drives\Drive X\Initialization\Init Config	
Init Speed	Definiert die Geschwindigkeit, mit welcher sich der Läufer bei der Initialisierung bewegt.
Home Position	Nachdem beim Initialisieren der mit dem Parameter **Maximal Init Current** vorgegebene Strom erreicht wurde, gilt der Motor als initialisiert. Dieser Position wird nun der Wert **Home Position** zugewiesen.
Initial Position	Am Ende der Initialisierung wird die **Initial Position** angefahren. Hat der Läufer diese Position erreicht, ist die Initialisierung abgeschlossen.



Generierung der Sollwerte

Run Mode

Der Modus der Sollwertvorgabe wird mit dem **Run Mode**-Parameter definiert. Dabei können die folgenden Modi unterschieden werden:

\Drives\Drive X\Set Value Generation\Run Mode	
Serial	Der Sollwert wird mit einem Protokoll über die serielle Schnittstelle vorgegeben.
Analog	Der Sollwert wird über den dem Motor entsprechenden analogen Eingang vorgegeben. Im Verzeichnis **Set Value Generation** können die Randsollwerte für die Eingangspegel 0V und 10V festgelegt werden. Der analoge Eingangsbereich zwischen 0V und 10V wird linear auf den Sollwertbereich abgebildet, der durch diese Randsollwerte (Maximal-/Minimalwert) definiert ist.
Continuous Curve	Eine auf dem Servo Controller abgespeicherte Sollwertkurve wird zyklisch abgefahren. Die Kurvennummer wird unter **Curve Number** im Verzeichnis **Set Value Configuration** gewählt.
Trigger Curve	Auf die steigende Flanke des Triggersignals wird eine erste, auf die fallende Flanke eine zweite auf dem Servo Controller gespeicherte Kurve abgefahren. Die Kurvennummern werden im Verzeichnis **Set Value Generation** unter **Rise Curve Number** und **Fall Curve Number** festgelegt. Folgt die fallende Flanke des Triggersignals noch bevor die erste Kurve zu Ende ist, wird die zweite Kurve unmittelbar anschliessend abgefahren.
Two Point	Ist das dem Antrieb entsprechende Triggersignal high, wird der Wert 'angefahren', welcher mit dem Parameter **'1' Position** bzw. **'1' Current** im Verzeichnis **Set Value Generation** definiert ist. Ist das Triggersignal low, wird der Wert 'angefahren', welcher mit dem Parameter **'0' Position** bzw. **'0' Current** definiert werden kann.

Set Value Configuration

Im Verzeichnis **Set Value Configuration** sind alle Konfigurations-Parameter zusammengefasst, die im Umgang mit der Sollwertvorgabe benötigt werden.

\Drives\Drive X\Set Value Generation\Set Value Configuration	
Mimimal Position L	Bestimmt die minimale Ist-Position (Untergrenze vom Positionierbereich).
Maximal Position L	Bestimmt die maximale Ist-Position (Obergrenze vom Positionierbereich).
0V Position L [1]	Dieser Parameter definiert im Modus **Analog** die Position, welche bei einer Eingangsspannung von 0 V angefahren werden soll.
10V Position L [1]	Dieser Parameter definiert im Modus **Analog** die Position, welche bei einer Eingangsspannung von 10 V angefahren werden soll.
'0' Position L [2]	Definiert die Position, welche im Modus **Two Point** angefahren wird, wenn der Trigger auf einer logischen Null liegt.
'1' Position L [2]	Definiert die Position, welche im Modues **Two Point** angefahren wird, wenn der Trigger auf einer logischen Eins liegt.
Curve Number L [3]	Im Modus **Continuous Curve** wird diejenige auf dem Servo Controller gespeicherte Kurve zyklisch 'abgefahren', welche die in **Curve Number** aufgeführte Nummer aufweist.
Rise Curve Number L [4]	Im Modus **Trigger Curve** wird die Kurve mit dieser Nummer auf die steigende Flanke des Triggersignals abgefahren.
Fall Curve Number L [4]	Im Modus **Trigger Curve** wird die Kurve mit dieser Nummer auf die fallende Flanke des Triggersignals abgefahren.
Curve Pos Offset L	Mit diesem Parameter kann der Positionsoffset der Kurve bestimmt werden.
Curve Amplitude L	Mit diesem Parameter kann die Amplitude der Kurve bestimmt werden. Der Wertebereich geht von 0 bis 100%.
Curve Speed L	Mit diesem Parameter kann die Geschwindigkeit der Kurve bestimmt werden. Der Wertebereich geht von 0 bis 100%.

1) Nur sichtbar im Modus **Analog**
2) Nur sichtbar im Modus **Two Point**
3) Nur sichtbar im Modus **Continuous Curve**
4) Nur sichtbar im Modus **Trigger Curve**

Die Abbildung 6-1, “Sollposition bei analoger Sollwertvorgabe”, auf Seite 79 zeigt wie sich der Sollwert im **Analog**-Modus verhält.



Filter Parameter

In diesem Verzeichnis werden alle Parameter aufgeführt, die für die Sollwert-Filterung notwendig sind.

\Drives\Drive X\Set Value Generation\Filter Parameter	
Max Speed L	Dieser Wert setzt die obere Grenze für die Geschwindigkeit. Er wird nicht berücksichtigt, wenn eine Kurve abgefahren wird! Beim Abfahren einer Kurve wird die Geschwindikeit und Beschleunigung aus der Kurve entnommen.
Max Acceleration L	Dieser Wert setzt eine obere Grenze für die Beschleunigung. Er wird nicht berücksichtigt, wenn eine Kurve abgefahren wird! Beim Abfahren einer Kurve wird die Geschwindikeit und Beschleunigung aus der Kurve entnommen.

Positionsüberwachung

Position Monitoring

Die Basissoftware unterstützt zwei Arten von Positionsüberwachung die im Kapitel “Positionsüberwachung” auf Seite 80 erklärt werden. Im Gegensatz zum Linearmotor ist es beim Schrittmotor nicht möglich einen Schleppfehler zu berechnen da die Positionsrückführung fehlt.

Folgende Parameter dienen zur Spezifikation der Schranken:

\Drives\Drive X\Position Monitoring L	
Pos Range Min **Pos Range Max**	Definiert die obere resp. untere Schranke der Positionsbandüberwachung. Liegt die Ist-Position des Antriebs unterhalb dieses Wertes, wird der digitale Ausgang **POSITION ERROR OUT** aktiviert.
In Position - **In Position +**	Diese zwei Parameter definieren wie nahe sich der Motor an der gewünschten Zielposition befinden muss, damit die Zielposition als erreicht gilt. Diese Parameter werden zur Zeit nur von der Multitriggersoftware benutzt. Wird einem Motor bei einem Zustandswechsel eine neue Sollposition vorgegeben (Relativ- oder Absolutbewegung) oder eine eine Kurve gestartet, geht das **In Position**-Signal des betreffenden Motors so lange auf logisch 0, bis sich der Läufer des Motors in einem Bereich um den Sollwert befindet.

Regel- und Steuer-Parameter

Control Parameters

In diesem Verzeichnis kann der Strom eingestellt werden mit dem der Schrittmotor betrieben wird.

\Drives\Drive X\Control Parameters	
Maximal Current	Bestimmt den Strom mit dem der Schrittmotor betrieben wird. Wenn der Schrittmotor stillsteht wird der Strom automatisch auf die Hälfte abgesenkt.



Regel- und Steuermodi

Control Switches

Dieser Parameter ist nur bei Servo Controllern der E100-Serie sichtbar und bestimmt ob mit 2 oder mit 3 Ampère Maximalstrom gearbeitet werden soll. Mit dem Parameter **Maximal Current** im Verzeichnis Control Parameter kann der Maximalstrom dann noch weiter eingegrenzt werden.

\Drives\Drive X\Control Switches	
Current ( ) 2A (x) 3A [1]	Wenn dieser Parameter selektiert ist, wird der Maximalstrom auf 3 Ampère ansonsten auf 2 Ampère begrenzt.

1) Nur sichtbar bei den Servo Controllern der Serie E100

Kommutierung

Commutation

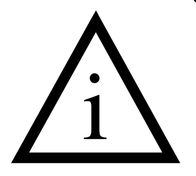
In diesem Verzeichnis sind jene Parameter aufgeführt, welche die Kommutierung der angeschlossenen Antriebe bestimmen. Die Kommutierung bestimmt wie die zwei Phasen des angeschlossenen Antriebes bestromt werden. Bei den Schrittmotoren hat die Kommutierung einen Einfluss auf die folgenden Betriebsgrössen:

- Auflösung
- Laufruhe
- maximal erreichbare Geschwindigkeit

\Drives\Drive X\Advanced\Commutation	
Auto (Default)	Auflösung: 4 Viertelschritte / Schritt max. Geschwindigkeit: Gross Bei dieser Kommutierung wird automatisch, in Abhängigkeit der Geschwindigkeit, zwischen den unten erklärten Kommutierungen hin- und hergeschaltet. Der Vorteil ist die hohe maximale Geschwindigkeit kombiniert mit der guten Auflösung.
Micro Step	Auflösung: 4 Viertelschritte / Schritt max. Geschwindigkeit: Klein Der Vorteil bei dieser Kommutierung ist die erreichbare Auflösung und der vibrationsarme Lauf. Der Nachteil ist die kleine maximal erreichbare Geschwindgkeit.
Half Step	Auflösung: 2 Viertelschritte / Schritt max. Geschwindigkeit: Mittel
Full Step	Auflösung: 1 Viertelschritte / Schritt max. Geschwindigkeit: Gross Der Vorteil bei dieser Kommutierung ist die maximal erreichbare Geschwindigkeit. Dem gegenüber steht die geringe Auflösung.



Stromauflösung



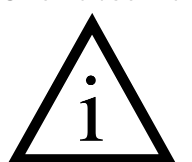
Da die Stromtreiber in den Geräten nur eine begrenzte Auflösung aufweisen, sollten nur die unten aufgelisteten Kombinationen von Maximalstrom (Phasenstrom) und Kommutierung eingestellt werden.

Kommutierung	Maximalstrom			
	LinMot E100/E200/E400[1]		LinMot E1000/ E2000/ E4000	LinMot E1001/ E2001/ E4001
	2 A	3 A		
Auto/Micro Step	1 / 2 A	1.5 / 3 A	0.75 / 1.5 / … / 6 A	0.75 / 1.5 / … / 7.5 A
Half Step	0.5 / 1 / 1.5 / 2 A	0.75 / 1.5 / 2.25 / 3 A	0.375 / 0.75 / … / 6 A	0.375 / 0.75 / … / 7.875 A
Full Step	> 0.125 A	> 0.178 A	> 0.05 A	> 0.05 A

Tabelle 6-4: Einstellung des Maximalstroms bei Schrittmotoren

1) Bei allen E100/E200/E400 Geräten mit Seriennummern grösser als ----.001.200 gilt nur die 3 A Kolonne!

Stromabsenkung



Führt der Rotor des Schrittmotors keine Bewegung aus, wird der Phasenstrom auf die Hälfte reduziert. Diese Reduktion des Phasenstroms geschieht 200 ms nach dem Stillstand automatisch. Der Parameter führ den Maximalstrom befindet sich im Verzeichnis **\Drives\Drive X\Control Parameters**.

Fehlerbehandlung

In diesem Unterkapitel wird die antriebsseitige Fehlerbehandlung beschrieben. Allgemeine Informationen zu Fehlern und Warnungen können dem Kapitel “Fehler-Behandlung” auf Seite 64 entnommen werden.

Error Mask
Warn Mask

In den unten aufgeführten Verzeichnissen kann für die Antriebsseite angegeben werden, bei welchen internen Fehlern in den Fehlerzustand gegangen wird (**Error Mask**), bei welchen Fehlern eine Warnung generiert wird (**Warn Mask**), wie die Temperaturüberwachung der Antriebe konfiguriert ist (**Temperature**) und was im Not-Stop-Fall geschehen soll (**Emergency Stop**).

Der Anwender kann nicht auf alle Error- und Warnparametern zugreifen, da bei fehlerhaften Einstellungen die Hardware Schaden nehmen könnte.

\Drives\Drive X\Error Handling\Error Mask	
Drive Type Mismatch R	Der angeschlossene Motorentyp stimmt nicht mit dem gewählten Motorentyp überein, oder der angeschlossene Motor ist defekt.
Curve Error R	Eine Sollwertkurve kann nicht gefunden werden.

Der Parameter Drive Init Not Done kann nicht deselektiert werden damit der Servo Controller nicht in den Zustand **RUN** gelangen kann, wenn ein Motor noch nicht initialisiert ist.

\Drives\Drive X\Error Handling\Warn Mask	
Drive Init Not Done R	Ein Motor hat nicht oder nicht korrekt initialisiert.
Pos Range Indicator	Der Läufer liegt ausserhalb des durch **Pos Range Min/Max** definierten Positionsbereichs. Wenn diese Warnung aktiviert wird muss darauf geachtet werden, dass die **Pos Range** Parameter so eingestellt werden, dass nach der Initialisierung das Signal nicht anliegt. Ansonsten kann nur schwer festgestellt werden ob alle Motoren korrekt initialisiert wurden. Der Ausgang wird bei einem AT Servo Controller nur aktiviert wenn auch der Parameter **Pos Range** im Verzeichnis **\System\IO Configuration** aktiviert ist. Bei einem MT Controller im MT-Modus (Parameter **MT** im Verzeichnis **\System\Command Interface** aktiviert) wird das Signal nur ausgegeben wenn der Parameter **Pos Range** im Verzeichnis **\Multi Trigger\Output Configuration\Output 3** aktiviert ist.
In Position	Ein Motor hat seine Sollposition erreicht. Der Ausgang **Msg Output** wird nur aktiviert wenn auch der Parameter **Msg Output** im Verzeichnis **\System\IO Configuration** aktiviert ist und der Parameter **In Position** im Verzeichnis **\System\Error Handling\Msg Mask** gesetzt ist.

Emergency Stop Mode

Dieser Parameter bestimmt, wie sich der Motor verhalten soll, nachdem der Eingang **STOP** aktiviert wurde.

\Drives\Drive X\Error Handling\Emergency Stop\Emergency Stop Mode	
Off	Der Motor wird nicht mehr angesteuert. Die Position wird weiterhin eingelesen. In diesem Modus werden die Wicklungen nicht mehr bestromt.
Freeze	Der Motor wird abgebremst[1] und in dieser Position gehalten.
Goto Position	Der Motor fährt[1] in die **Emergency Stop Position** und wird auf diese Position geregelt.

1) Die Geschwindigkeit und Beschleunigung dieser Bewegungen kann mit den Parametern **Max Speed** und **Max Acceleration** im Verzeichnis **Emergency Configuration** eingestellt werden



Emergency Configuration

Die Parameter in diesem Verzeichnis dienen zur Parametrierung des Fahrverhaltens beim Aktivieren des Eingangs **STOP**.

\Drives\Drive X\Error Handling\Emergency Stop\Emergency Configuration [1]	
Stop Position L [2]	Ist der Notaus-Modus **Drive Goto Position** selektiert, wird beim Aktivieren des Eingangs **STOP** unverzüglich die unter **Emergency Stop Position** definierte Position angefahren. Der Motor bleibt in Betrieb.
Max Speed L	Bestimmt, wie schnell sich der Antrieb in die **Stop Position** bewegt.
Max Acceleration L	Bestimmt, mit welcher Beschleunigung der Antrieb beim Aktivieren des Eingangs **STOP** abgebremst wird.

1) Nur sichtbar, wenn der Parameter **Emergency Stop Mode** auf **Goto Position** oder **Freeze** eingestellt ist.
2) Nur sichtbar wenn der Parameter **Emergency Stop Mode** auf **Goto Position** eingestellt ist.

6.6 Magnet-Parameter

In diesem Kapitel werden alle Parameter erklärt welche für die Parametrierung von Magneten benötigt werden. Damit diese Parameter im Parameterinspektor sichtbar werden, muss im Verzeichnis **\Drives\Drive X\Type** der Parameter **Magnet** gewählt werden.

Master / Booster

Dieser Parameter muss in der Betriebsart **Magnet** auf **Master** eingestellt werden.

\Drives\Drive X\Advanced\Master / Booster	
Master	Der angeschlossene Aktor ist ein Master.

Generierung der Sollwerte

In diesem Verzeichnis wird die Art der Sollwertvorgabe eingestellt.

Run Mode

Der Modus der Sollwertvorgabe wird mit dem **Run Mode**-Parameter definiert. Dabei können die folgenden Modi unterschieden werden:

\Drives\Drive X\Set Value Generation\Run Mode	
Serial	Der Sollwert wird mit einem Protokoll über die serielle Schnittstelle vorgegeben. Das kundenspezifische Protokoll wird in der "Customized Application Software" realisiert.
Analog	Der Sollwert wird über den dem Motor entsprechenden analogen Eingang vorgegeben. Im Verzeichnis **Set Value Generation** können die Randsollwerte für die Eingangspegel 0V und 10V festgelegt werden. Der analoge Eingangsbereich zwischen 0V und 10V wird linear auf den Sollwertbereich abgebildet, der durch diese Randsollwerte (Maximal-/Minimalwert) definiert ist.
Continuous Curve	Eine auf dem Servo Controller abgespeicherte Sollwertkurve wird zyklisch abgefahren. Die Kurvennummer wird unter **Curve Number** im Verzeichnis **Set Value Configuration** gewählt.
Trigger Curve	Auf die steigende Flanke des Triggersignals wird eine erste, auf die fallende Flanke eine zweite auf dem Servo Controller gespeicherte Stromkurve abgefahren. Die Kurvennummern werden im Verzeichnis **Set Value Generation** unter **Rise Curve Number** und **Fall Curve Number** festgelegt. Folgt die fallende Flanke des Triggersignals noch bevor die erste Kurve zu Ende ist, wird die zweite Kurve unmittelbar anschliessend abgefahren.
Two Point	Ist das dem Antrieb entsprechende Triggersignal high, wird der Strom eingestellt, welcher mit dem Parameter **'1' Current** im Verzeichnis **Set Value Generation** definiert ist. Ist das Triggersignal low, wird der Strom eingestellt, welcher mit dem Parameter **'0' Current** definiert werden kann.



Set Value Configuration

Im Verzeichnis **Set Value Configuration** sind alle Konfigurations-Parameter zusammengefasst, die im Umgang mit der Sollwertvorgabe benötigt werden.

\Drives\Drive X\Set Value Generation\Set Value Configuration	
Minimal Current L	Bestimmt den minimalen Ist-Strom (Untergrenze vom Strombereich).
Maximal Current L	Bestimmt den maximalen Ist-Strom (Obergrenze vom Strombereich).
0V Current L [1]	Bei Ansteuerung eines Magneten bestimmt dieser Parameter im Modus **Analog** den Strom, welcher bei einer Eingangsspannung von 0V, durch die Wicklung fliessen soll.
10V Current L [1]	Bei Ansteuerung eines Magneten bestimmt dieser Parameter im Modus **Analog** den Strom, welcher bei einer Eingangsspannung von 10V, durch die Wicklung fliessen soll.
‘0’ Current L [2]	Beim Magneten definiert dieser Wert den Strom, welcher bei einer logischen Null des Triggers getrieben wird.
‘1’ Current L [2]	Beim Magneten definiert dieser Wert den Strom, welcher bei einer logischen Eins des Triggers getrieben wird.
Curve Number L [3]	Im Modus **Continuous Curve** wird diejenige auf dem Servo Controller gespeicherte Kurve zyklisch 'abgefahren', welche die in **Curve Number** aufgeführte Nummer aufweist.
Rise Curve Number L [4]	Im Modus **Trigger Curve** wird die Kurve mit dieser Nummer auf die steigende Flanke des Triggersignals abgefahren.
Fall Curve Number L [4]	Im Modus **Trigger Curve** wird die Kurve mit dieser Nummer auf die fallende Flanke des Triggersignals abgefahren.
Curve Current Offset L	Mit diesem Parameter kann der Stromoffset der Kurve bestimmt werden.
Curve Amplitude L	Mit diesem Parameter kann die Amplitude der Kurve bestimmt werden. Der Wertebereich geht von 0 bis 100%.
Curve Speed L	Mit diesem Parameter kann die Geschwindigkeit der Kurve bestimmt werden. Der Wertebereich geht von 0 bis 100%.

1) Nur sichtbar im Modus **Analog**
2) Nur sichtbar im Modus **Two Point**
3) Nur sichtbar im Modus **Continuous Curve**
4) Nur sichtbar im Modus **Trigger Curve**



Die folgende Grafik verdeutlicht die Abbildung der analogen Eingangsspannung auf den Sollstrom im **Analog**-Modus.

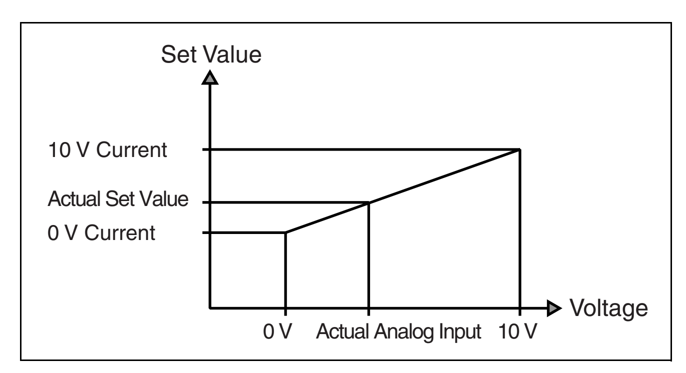


Abbildung 6-2: Sollstrom bei analoger Sollwertvorgabe

Der errechnete Sollwert ist der zwischen dem Parameter **0V Current** und **10V Current** linear interpolierte Wert.

Regel- und Steuer-Parameter

Control Parameters

In diesem Verzeichnis sind jene Parameter aufgeführt, welche bei Steuerung der Magnete benötigt werden.

\Drives\Drive X\Control Parameters	
Maximal Current L	Bestimmt den maximalen Strom der vom Regler eingestellt werden kann. Dieser Parameter bestimmt durch den maximalen Strom auch die maximale Kraft.

Regel- und Steuermodi

Control Switches

Dieser Parameter ist nur bei Servo Controllern der E100-Serie sichtbar und bestimmt ob mit 2 oder mit 3 Ampère Maximalstrom gearbeitet werden soll. Mit dem Parameter **Maximal Current** im Verzeichnis Control Parameter kann der Maximalstrom dann noch weiter eingegrenzt werden.

\Drives\Drive X\Control Switches	
Current ( ) 2A (x) 3A [1]	Wenn dieser Parameter selektiert ist, wird der Maximalstrom auf 3 Ampère ansonsten auf 2 Ampère begrenzt.

1) Nur sichtbar bei den Servo Controllern der Serie E100



Fehlerbehandlung

In diesem Unterkapitel wird die antriebsseitige Fehlerbehandlung beschrieben. Allgemeine Informationen zu Fehlern und Warnungen können dem Kapitel “Fehler-Behandlung” auf Seite 64 entnommen werden.

Error Mask

In den unten aufgeführten Verzeichnissen kann für die Antriebsseite angegeben werden, bei welchen internen Fehlern in den Fehlerzustand gegangen wird (**Error Mask**) und was im Not-Stop-Fall geschehen soll (**Emergency Stop**).

\Drives\Drive X\Error Handling\Error Mask	
Drive Type Mismatch R	Der angeschlossene Motorentyp stimmt nicht mit dem gewählten Motorentyp überein, oder der angeschlossene Motor ist defekt.
Curve Error R	Eine Sollwertkurve kann nicht gefunden werden.

Emergency Stop Mode

Mit dem folgenen Modus kann das Verhalten des Magnets nach einem Notaus definiert werden.

\Drives\Drive X\Error Handling\Emergency Stop\Emergency Stop Mode	
Off	Der Magnet wird nicht mehr bestromt.
Freeze	Der beim Auftreten des Not-Stops aktuelle Stromwert wird ‘eingefroren’. Der Magnet bleibt in Betrieb.
Set Current	Beim Auftreten eines Not-Stops wird unverzüglich der unter **Stop Current** definierte Strom getrieben.

Emergency Configuration

Dieses Verzeichnis enthält nur einen Parameter der definiert welcher Strom bei einem Notaus in den Magnet getrieben wird.

\Drives\Drive X\Error Handling\Emergency Stop\Emergency Configuration [1]	
Stop Current L	Ist der **Set Current** Notaus-Modus selektiert, wird beim Auftreten eines Notaus unverzüglich dieser Strom in den angeschlossenen Magneten getrieben.

1) Nur sichtbar wenn der Parameter **Emergency Stop Mode** auf **Set Current** eingestellt ist.

6.7 Positionssensorik-Parameter

Mit Hilfe der externen Sensorik ist es möglich, die Wiederholgenauigkeit sowie die Linearität der *LinMot*® Motoren zu erhöhen. Es werden Sinus/Cosinus-Geber mit einem Polabstand von 1 mm und 5 mm unterstützt. Die externe Sensorik kann mit zwei Parametern konfiguriert werden. Im Kapitel “Externe Positionssensorik” auf Seite 54 wird die Thematik genauer erläutert.

Sensor Period

Mit diesem Parameter wird der Polabstand des angeschlossenen Gebers eingestellt. Es kann entweder 1 mm oder 5 mm gewählt werden.

\Drives\Drive X\Sensor Configuration\Sensor Period	
1 mm	Muss angewählt werden, wenn die angeschlossene Sensorik einen Polabstand von 1 mm hat.
5 mm	Muss angewählt werden, wenn die angeschlossene Sensorik einen Polabstand von 5 mm hat.

Sensor Direction

Mit diesem Parameter wird die Richtung des Sensors gegenüber der Richtung des Motors eingestellt.

\Drives\Drive X\Sensor Configuration\Sensor Direction	
Positive	Muss eingestellt werden, wenn bei einer Bewegung des Läufers aus dem Motor heraus das Sinus- vor dem Cosinus-Signal kommt.
Negative	Muss eingestellt werden, wenn bei einer Bewegung des Läufers aus dem Motor heraus das Cosinus- vor dem Sinus-Signal kommt.

Error Mask Warn Mask

Das untenstehende Verzeichnis zeigt an, bei welchen Fehlern in den Fehlerzustand gewechselt wird (**Error Mask**) und wleche Warnungen auftreten können (**Warn Mask**).

\Drives\Drive X\Error Handling\Error Mask	
Drive Type Mismatch R	Der angeschlossene Sensor entspricht nicht dem konfigurierten oder der Sensor ist defekt.
Slider Missing R	Der Sensor fehlt, nicht korrekt montiert oder nicht korrekt angeschlossen.

\Drives\Drive X\Error Handling\Warn Mask	
Slider Missing	Der Sensor ist nicht korrekt montiert oder angeschlossen.



6.8 MT-Parameter

Die Parameter für die MT Servo Controller dienen dazu die Ein- und Ausgänge zu konfigurieren.

Laufzeitstreuung

Jitter Filter

Über die vier digitalen Eingänge können bis zu 16 Zustände definiert werden, die von der übergeordneten Steuerung einzeln aufgerufen werden können. Sobald bei einem Zustandswechsel mehr als ein digitales Eingangssignal seinen Zustand wechselt, kann es zu Laufzeitstreuungseffekten kommen.

Wird beispielsweise vom Zustand 0 (Eingangskombination 0000) der Zustand 3 (0011) aufgerufen, sollten die zwei letzten Eingangsbits zum exakt gleichen Zeitpunkt wechseln. Ist dies nicht der Fall, wird während dem Wechsel kurzzeitig der Zustand 1 (0001) bzw. der Zustand 2 (0010) eingenommen. Wenn nun der Servo Controller diese transienten Zustände erkennen würde, hätte dies unerwünschte Konsequenzen. Um dies zu verhindern, wird ein neuer Zustand erst eingenommen, nachdem die Eingangssignale während einer einstellbaren Zeitdauer stabil anliegen.

Multi Trigger\Jitter Filter	
Time	Zeitdauer während der die Eingangssignale stabil anliegen müssen, damit ein Zustandswechsel ausgeführt wird.

Bestätigung

Acknowledge

Mit diesem Parameter kann bestimmt werden wie lange das Signal **In Position** mindestens auf null geht, nachdem ein Kommando ausgeführt wurde. Das Signal **In Position** geht erst auf eins wenn diese Zeitdauer abgelaufen ist **und** der Motor an der gewünschten Position ist.

Multi Trigger\Acknowledge	
Time	Minimale Zeitdauer während der das Signal **In Position** auf Null geht nachdem ein Kommando ausgeführt wurde.

Ausgangssignale

Output Configuration

Die Ausgänge 3 und 4 lassen sich vom Anwender mittels der Parameter im Verzeichnis **Multi Trigger\Output Configuration** konfigurieren.

Output 3
Output 4

Mit Hilfe dieses Parameters kann die Funktion des Ausgangs 3 und 4 eingestellt werden.

Multi Trigger\Output Configuration\Output 3	
None	Ausgang wird nicht angesteuert
In Pos A /In Pos B **In Pos C /In Pos D**	**In Position**-Signal von Motor A, B, C und D
In Pos A*B	UND-Verknüpfung der jeweiligen **In Position**-Signale
In Pos C*D	
In Pos A*B*C*D	
Pos Range	Das **Pos Range**-Signal wird ausgegeben

Multi Trigger\Output Configuration\Output 4	
None	Ausgang wird nicht angesteuert
In Pos A /In Pos B **In Pos C /In Pos D**	**In Position**-Signal von Motor A, B, C und D
In Pos A*B	UND-Verknüpfung der jeweiligen **In Position**-Signale
In Pos C*D	
In Pos A*B*C*D	



6.9 PROFIBUS-Parameter

In diesem Kapitel werden die Parameter der PROFIBUS-DP Servo Controller erklärt.

Diagnose Priority

Mit Hilfe des Parameters **Diagnose Priority** legt der Anwender fest, mit welcher Priorität der *LinMot*® Servo Controller im Fehlerfall die Diagnose von der SPS anfordert.

PROFIBUS-DP\Diagnose Priority	
None	Es wird keine Diagnose von der SPS angefordert.
Low	Die Diagnose wird mit tiefer Priorität von der SPS angefordert. Der zyklische Programm (OB1) der SPS wird dadurch nicht unterbrochen.
High	Die Diagnose wird mit hoher Priorität von der SPS angefordert. Das zyklische Programm (OB1) der SPS wird durch einen hochpriorisierten OB unterbrochen.

Byte Order Datamodules

Der Parameter **Byte Order Datamodules** legt die Byte-Reihenfolge fest, mit welcher die Datenmodule auf dem *LinMot*® Servo Controller ausgewertet und versendet werden.

PROFIBUS-DP\Byte Order Datamodules	
Reversed	Wenn der Parameter angewählt ist, wird die Byte-Reihenfolge der Datenmodule invertiert. Bei Siemens SPS-Steuerungen sollte dieser Parameter angewählt werden.

Info

Die im Verzeichnis **Info** angezeigten Parameter sind nur gültig, wenn der Servo Controller gestartet ist.

PROFIBUS-DP\Info	
Slave Node Address	Zeigt die PROFIBUS-Adresse des *LinMot*® Servo Controllers.
Master Node Address	Zeigt die PROFIBUS-Adresse des Masters.
Baudrate	Zeigt die erkannte Baudrate an. Dieser Parameter wird nur beschrieben, falls der Servo Controller korrekt vom DP-Master eingebunden wird und sich im DataExchange-Zustand befindet.
Bus Cycle Time	Zeigt die aktuell auf dem Kontroller gemessene PRFIBUS Zykluszeit an.Servo Controller.



Interface Card Type

Mit diesem Parameter wird der **Interface Card Type** gewählt.

Interface Card Type	
None	Keine Interfacekarte ist angeschlossen oder benutzt.
DI01-08/08	Ein digitales I/O Module DI01-08/08 is angeschlossen, mit 8 digitalen Eingängen und 8 digitalen Ausgängen.
ME01-01/08	Ein Masterencoder-Modul ME01-01/08 ist angeschlossen mit einem Encoder Link, 8 digitalen Eingängen und 8 digitalen Ausgängen.
ME01-02/08	Ein Masterencoder-Modul ME01-02/08 ist angeschlossen mit zwei Encoder Links, 8 digitalen Eingängen und 8 digitalen Ausgängen.



7. Tips und Tricks zum Regler

7.1 Einführung

Dieses Kapitel vermittelt ohne weitergehende theoretische Erläuterungen das für die Praxis notwendige Wissen zur Einstellung des Positionsreglers in den *LinMot®* Servo Controllern. Dem theoretisch interessierten Anwender empfehlen wir die einschlägige Literatur zu konsultieren.

Die folgende Abbildung zeigt das Reglerschema. Die kursiv geschriebenen Bezeichnungen stehen für einstellbare Parameter. Grundsätzlich handelt es sich um einen PID-Regler mit Feed-Forward Strukturen. Zusätzlich steht ein v/a-Limiter für die Vorfilterung der Sollposition bzw. eine Sollwertvorgabe über Kurvenprofile zur Verfügung.

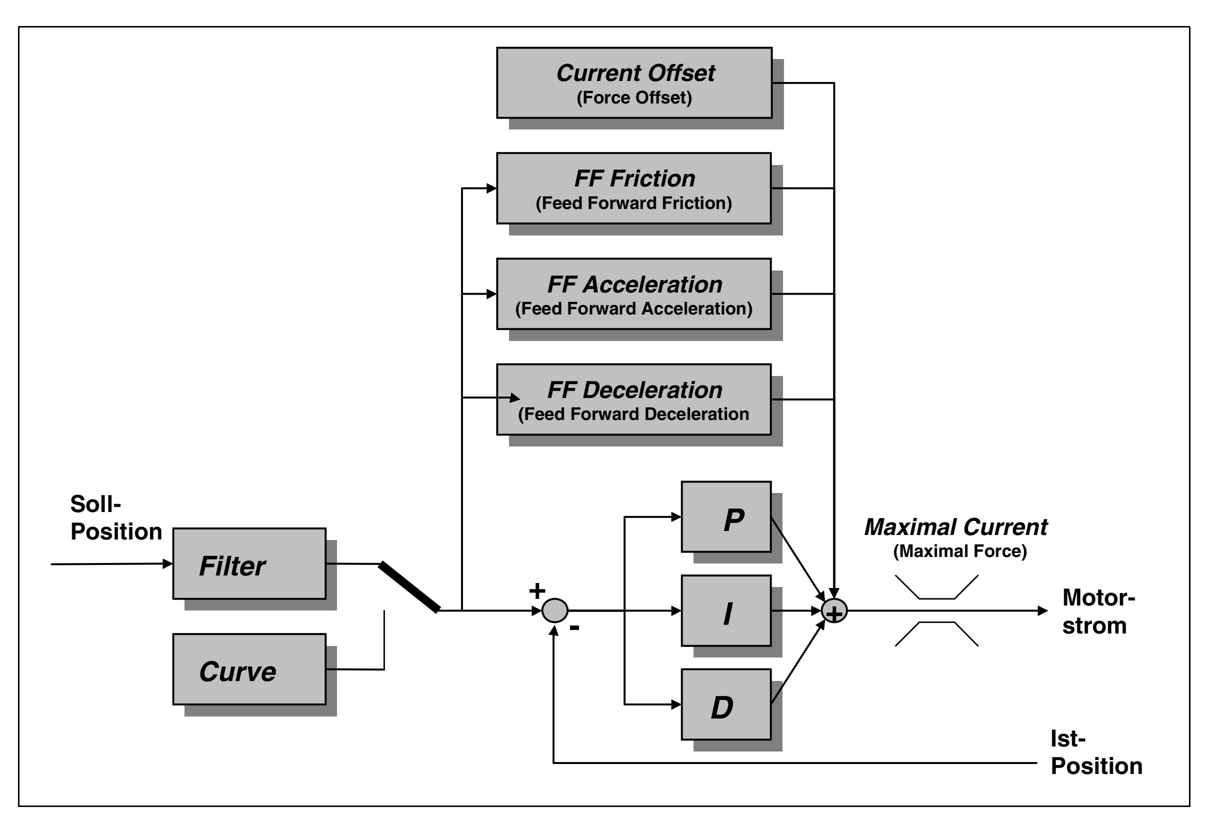


Abbildung 7-1: Reglerschema

Bei langsamen Bewegungen mit geringen Lastmassen genügen oftmals die Default-Einstellungen der Reglerparameter, sofern das Vorfilter bzw. die Kurvenprofile richtig gewählt wurden. Bei schnelleren Bewegungen und bei grossen Lastmassen ist konsequent nach den Einstellregeln (siehe nachfolgende Abschnitte) vorzugehen. Von einem ‘gefühlsmässigen’ Parametrieren ist strikte abzusehen!

7.2 Wahl zwischen PD- oder PID- Regler

PD-Regler

Ein PD-Regler ist ein Regler, der nur mit P- und D-Anteil arbeitet (I=0). Diese Art von Regler ist einfach einzustellen und zeigt ein sehr robustes Verhalten. Der Nachteil eines solchen Reglers ist, dass evt. vorhandene statische Positions-Abweichungen nicht automatisch korrigiert werden.

PID-Regler

Ein PID-Regler ist ein Regler, der mit P-, D- und I-Anteil arbeitet. Diese Art von Regler kann evt. vorhandene statische Reglerabweichungen dank dem I-Anteil korrigieren. Der Nachteil dieses Reglers ist, dass Systemschwingungen leister auftreten können. Bei der Einstellung des Reglers können die Werte vom PD-Regler übernommen werden und der I-Anteil langsam erhöht werden. Je höher der I-Anteil gesetzt wird, desto schneller wird der Regler evt. vorhandene Positionsabweichungen korrigieren. Ein zu grosser I-Wert kann aber zu einem instabilen Regelverhalten führen, insbesondere falls der Motor der Sollwertkurve teilweise nicht folgen kann.

7.3 Einstellung des Vorfilters (Filter)

Das Vorfilter beschränkt die max. Beschleunigung und Geschwindigkeit nach der Zielvorstellung des Anwenders. Damit kann von der übergeordneten Steuerung ein rechteckiger Sollwertsprung der Position vorgegeben werden und der Antrieb führt dennoch eine ‘sanfte’ in der max. Beschleunigung und Geschwindigkeit limitierte Bewegung aus. Grundsätzlich darf der Sollwert nie ‘schneller’ geändert werden, als dass der Motor aufgrund der zur Verfügung stehenden Kräfte der Bewegung folgen kann! <u>Der Einstellung des Vorfilters kommt eine höhere Bedeutung zu als der optimalen Einstellung der Reglerparameter !</u>

Die maximale Geschwindigkeit und Beschleunigung, die ein Motor erreichen kann, hängt in komplexer Weise mit den Anwendungsparametern (Massen, Reibung, Geschwindigkeit/Kraft-Verhältnis, ...) zusammen. Es wird daher dringend empfohlen, die reale Anwendung mittels des Programmes *LinMot*® Designer (siehe LinMot CD oder unter www.linmot.com) zu simulieren und daraus die entsprechenden Maximalwerte auszulesen.

Wird mit Sollwertkurven gearbeitet, entfällt die Funktion des Vorfilters bzw. die max. Geschwindigkeit und Beschleunigung des Motors müssen bei der Generierung der Sollwertkurve beachtet werden.

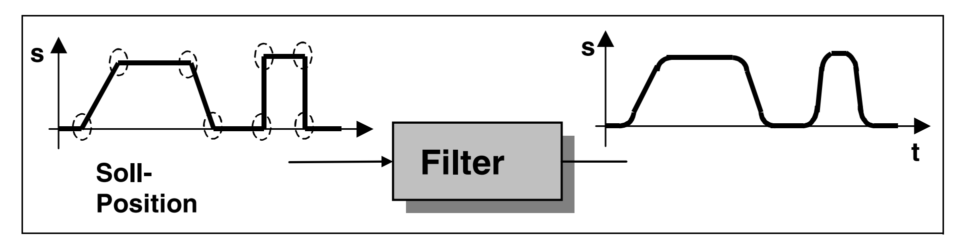


Abbildung 7-2: Sollwertsignal vor und nach dem Vorfilter



7.4 Sollwertvorgabe mittels Kurven

Wird eine Betriebsart gewählt, bei der die Sollwertvorgabe über Kurvenprofile vorgegeben wird, sind folgende vier Punkte zu beachten:

a) die Sollwertkurve muss vom Motor physikalisch realisierbar sein. Der Sollwert darf nie 'schneller' geändert werden, als dass der Motor aufgrund der zur Verfügung stehenden Kräfte der Bewegung folgen kann! Aus der Bewegungssimulation mit dem *LinMot®* Designer sind die entsprechenden Werte (Geschwindigkeit, Beschleunigung) für eine bestimme Applikation auszulesen.

b) die Bewegungskurven sollten in jedem Fall geglättet sein. Jedwelche Positionssprünge oder sprunghaften Aenderungen der Geschwindigkeit müssen eliminiert werden. Dazu stehen im Curve Inspector von *LinMot®* Talk die entsprechenden Kurvenfunktionen zur Verfügung (Bsp.: Sinuskurve). Insbesondere sollten keine Rampenfunktionen verwendet werden, da hier Geschwindigkeitssprünge auftreten!

c) Bei der Generierung der Sollwertkurve mittels des 'Curve Inspectors' von *LinMot®* Talk sollten nicht mehr Stützpunkte vorgegeben werden als vom Curve Inspector vorgeschlagen werden (gilt ab Release 1.3.9)! Hört sich die Bewegung 'grob und rauh' an, so ist die Anzahl Stützpunkte zu reduzieren!

d) Wird eine Kurve mittels eines externen Tools erzeugt, so soll als Regel für die Anzahl Stützpunkte gelten: minimal 16 Stützpunkte, wobei die minimale Distanz zwischen den Stützpunkten 1 ms betragen soll (eine 10 mm Hubbewegung ausgeführt in 14 ms sollte mit 14 Stützpunkten generiert werden). In allen anderen Fällen sollten Stützpunkte im ca 5 ms Raster generiert werden.

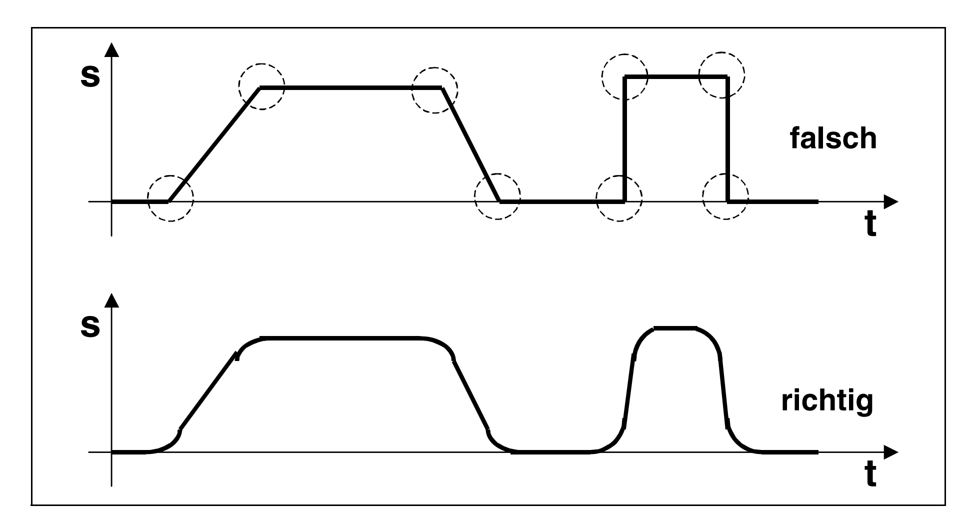


Abbildung 7-3: Kurvenprofile müssen 'geglättet' sein

7.5 Einstellung der Vorsteuerung (Feed-Forward)

Der Begriff ‘Vorsteuerung’ wird in der Regelungstechnik für das vorausschauende Steuern oder Berechnen einer Stellgrösse verwendet. Durch dieses ‘Vorausschauen’ kann der Regler wesentlich besser auf die Problemstellung eingestellt werden. Wenn der Regler ‘weiss’, dass im System eine grosse Reibung vorhanden ist, und er den Reibungskoeffizienten kennt, kann er bei einer gewünschten Bewegung den nötigen Strom vorsteuern, um diese Reibung zu kompensieren. Damit entsteht von Anfang an eine viel geringere Abweichung zwischen der Ist- und der Sollposition.

Der im *LinMot®* Servo Controller integrierte Regler kennt drei dieser Vorsteuerungsparameter, mit welchen sich das Regelverhalten und somit die erreichbare Dynamik verbessern lässt.

FF Friction

Mit Hilfe des Parameters **FF Friction** kann die Gleitreibung des Systems kompensiert werden. Dieser Wert kann folgendermassen berechnet werden:

FF Friction = F_{FR} / c_f		
	FF Friction	Vorsteuerung Reibung [A]
	F_{FR}	Gleitreibung [N]
	c_f	Kraftkonstante Motor [N/A]

Wobei F_{FR} der Gleitreibung und c_f der Kraftkonstante des gewünschten Motors entspricht.

Hinweis:
F_{FR} lässt sich mittels einer Federwaage bestimmen (Motor ausstecken) und der Wert für die Kraftkonstante c_f ist im Datenblatt des jeweiligen Motors aufgeführt. Bei langen Hüben kann auch der Motorstrom während der konstanten Bewegungsphase mit dem Oszilloskop ausgelesen werden und man erhält so direkt den Wert für FF Friction.

FF Acceleration

Der Parameter **FF Acceleration** wird dem Regler beim Beschleunigen helfen, indem proportional zur geforderten Beschleunigung ein Strom vorgesteuert wird. Dieser Parameter sollte überall dort verwendet werden, wo sehr schnelle und dynamische Bewegungen erforderlich sind, oder wenn grosse Lastmassen bewegt werden. Der Wert dieses Parameters kann folgendermassen berechnet werden:

FF Acceleration = (m / c_f)		
	FF Acc.	Vorsteuerung [mA/(m/s²)]
	m	Bewegte Masse [g]
	c_f	Kraftkonstante Motor [N/A]

Wobei m der bewegten Masse (Lastmasse + Masse des Läufers oder des Stators) und c_f der Kraftkonstanten des gewünschten Motors entspricht. Die Kraftkonstante c_f kann aus den Datenblättern des betreffenden Motors abgelesen werden.



FF Deceleration

Dieser Parameter ist das Gegenstück zum Parameter FF Acceleration und wird während dem Abbremsen des Motors zur Vorsteuerung des Stromes verwendet. Der Wert dieses Parameters kann folgendermassen berechnet werden:

FF Deceleration = (m / c_f)	FF Dec.	Vorsteuerung [mA/(m/s²)]
	m	Bewegte Masse [g]
	c_f	Kraftkonstante Motor [N/A]

Wobei m der bewegten Masse[1] (Lastmasse + Masse des Läufers oder des Stators) und c_f der Kraftkonstanten des gewünschten Motors entspricht. Die Kraftkonstante c_f kann aus den Datenblättern des betreffenden Motors abgelesen werden.

Beispiel

Ein Linearmotor P01-37x240/60x260 bewegt eine Lastmasse von 1200g auf einer Zusatzführung. Für die Gleitreibung wird eine Kraft von 10 N gemessen. Der Linearmotor hat gemäss Datenblatt eine Kraftkonstante c_f von 40 N/A. Das Läufergewicht beträgt 829 g. Dies ergibt zusammen mit der Läufermasse eine total bewegte Masse von 2029 g.

Durch Einsetzen in obenstehende Formeln erhält man:

FF Friction	0.25 A
FF Acceleration	50.7 mA/(m/s²)
FF Deceleration	50.7 mA/(m/s²)

7.6 Einstellung des Stromoffsets (Current-Offset)

Horizontale Bewegung

Bei Anwendungen mit horizontaler Bewegungsrichtung sind die Verhältnisse für Vor- und Rückwärtsbewegung identisch und der Parameter **Current Offset** muss Null sein.

Vertikale Bewegung

Bei Anwendungen mit vertikaler Bewegungsrichtung führt die Gewichtskraft zu einem unsymmetrischen Regelverhalten für die Auf- und die Abwärtsbewegung. Mit dem Parameter **Current Offset** im Verzeichnis **\Drives\Drive X\Control Parameters** kann diese Unsymmetrie ausgeglichen werden. Der Wert wird gemäss der folgenden Formel berechnet:

Offset Current = (m * g) / c_f	m	Lastmasse [kg]
	g	Erdbeschleunigung 9.81 m/s²
	c_f	Kraftkonstante Motor [N/A]

Die Masse m entspricht der bewegten Masse (Lastmasse und Läufer- oder Statormasse). Die Kraftkonstante c_f ist im Datenblatt des betreffenden Motors aufgeführt. Das Vorzeichen des Parameters **Current Offset** ist von der Einbaulage abhängig. Zeigt der Kabelaustritt des Motors zum Boden muss das positive ansonsten das negative Vorzeichen gewählt werden.

Beispiel

Ein Linearmotor P01-37x240/60x260 in vertikaler Einbaulage (Kabelaustritt nach unten) bewegt eine Lastmasse von 1.2 kg. Der Lineamotor hat gemäss Datenblatt eine Kraftkonstante c_f von 40 N/A und das Läufergewicht beträgt 0.829 kg. Dies ergibt zusammen mit der Last eine bewegte Masse von 2.029 kg. Gemäss obenstehender Formel ergibt sich ein Offset-Strom von 0.5 A.

Tip:

Sofern die Masse einer Applikation nicht bekannt ist, kann folgendermassen vorgegangen werden: Man hält den Motor an einer bestimmten Position an: Der dazu benötigte Motorstrom kann mit dem Oszilloskop ausgelesen werden oder man kann den max. Strom des Motors solange reduzieren, bis der Motor diese Position nicht mehr halten kann. Beide Ergebnisse des Stromwertes ergeben in etwa den benötigten Offsetstrom.



7.7 Das Tuning Tool

Das sog. Tuning Tool wurde mit dem Software-Release 1.3.10 eingeführt. Es hilft dem Benutzer die **Feed Forward Parameter** und den **Current Offset** zu berechnen und einzustellen, ohne dass die Motorendatenblätter begezogen werden müssen. without need of reading the motor data sheet. Das Tuning Tool wird mit der Taste “Show Tuning Tool” im Parameter Inspector gestartet..

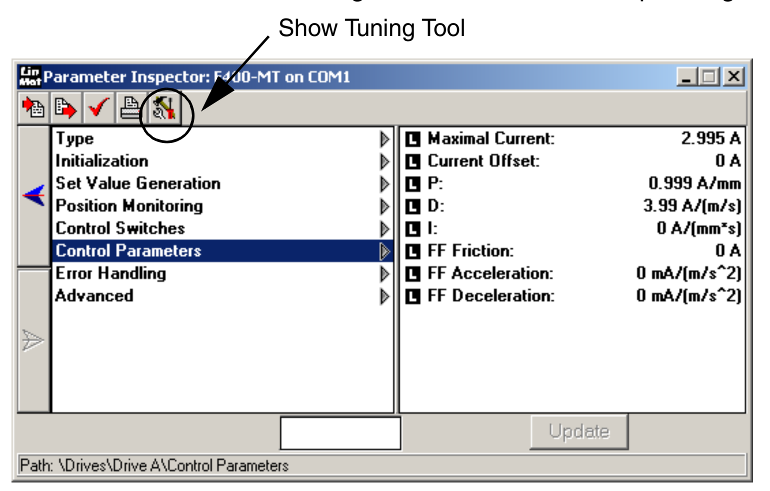


Abbildung 7-4: Parameter Inspector

Beispiel: Benutzen des Tuning Tools

Ein Linearmotor **P01-37x240/60x260** ist vertikal montiert (positive Richtung entgegen der Gravitationskraft) und bewegt mit dem Läufer eine Lastmasse von **1.2 kg**. Dieser Linearmotor hat eine Kraftkonstante c_f von 40.8 N/A und die Läufermasse ist 829 g. Dies ergibt eine bewegte Gesamtmasse von 2029 g.

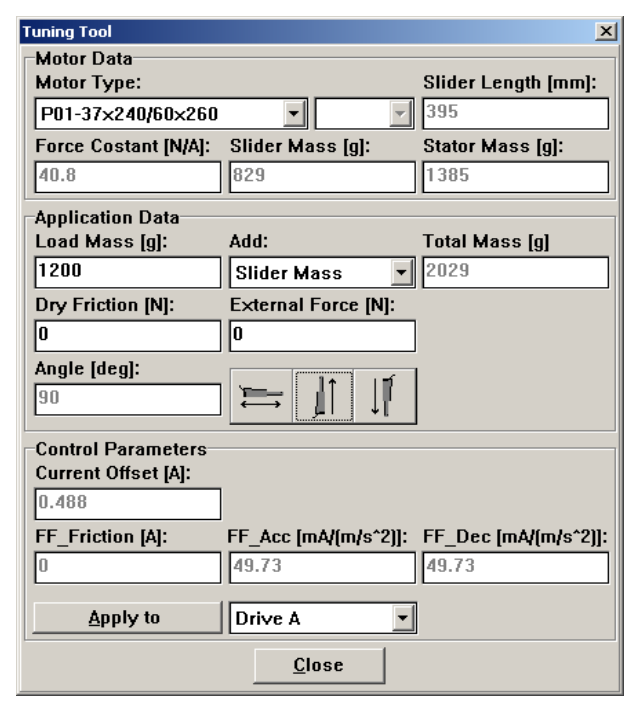


Abbildung 7-5: Das Tuning Tool



Motorendaten

Im PopUp “Motor Type” wird der Motorentyp ausgewählt. Spezialmotoren “F” (Fast-Motor) oder “S” (Short-Motor) werden mit dem rechten PopUp-Menu angegeben. Die Läuferlänge, Läufermasse, Statormasse und Kraftkonstante werden automatisch angezeigt.

Beachte:

Fall der gewünschte Motor nicht in der Liste enthalten ist, kann mit dem Typ “Other...” die Kraftkonstante, Läufermasse und Statorgewicht in die entsprechenden Felder eingegeben werden (in diesem Fall müssen die Werte dem Datenblatt entnommen werden).

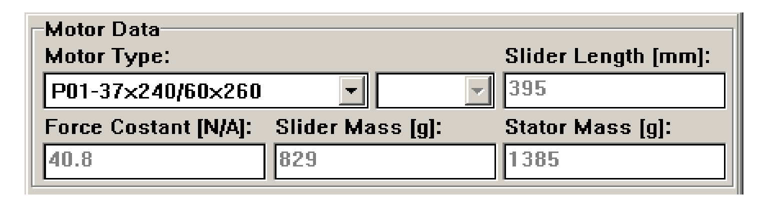


Abbildung 7-6: Auswahl des Motorentyps

Applikationsdaten

Setzen Sie die Lastmasse im Feld “Load Mass” ein und wählen mit dem “Add” PopUp den bewegten Teil des Motors aus. Die Masse des bewegten Teils wird zur Lastmasse hinzuaddiert. Die Gesamtmasse wird im Feld “Total Mass” angezeigt. Geben Sie die Gleitreibung im Feld “Dry Friction” ein. Falls eine zusätzliche Konstantkraft vorhanden ist (z.B. MagSpring), setzen Sie den entsprechenden Wert ins Feld “External Force”. Das Vorzeichen dieser Kraft ist positiv, wenn sie in positive Bewegungsrichtung gerichtet ist. Wählen Sie die Einbaulage des Motors mit der entsprechenden Taste aus. (siehe Abbildung 7-7, “Eingabe der Applikationsdaten”).

Beachte:

Falls mehrere Motoren parallel im Master/Booster oder Gantry-Modus arbeiten, sind die Lastmasse und die Masse aller bewegten Teile der Motoren zu addieren und durch die Anzahl Motoren zu dividieren. Setzen Sie die errechneten Werte in das Feld “Load Mass” und klicken Sie im PopUp “Add” auf “None”. Mit diesem Vorgehen koennen auch die FF Parameter und der Strom Offset für den Master Motor und alle Gantry-Motoren berechnet werden!

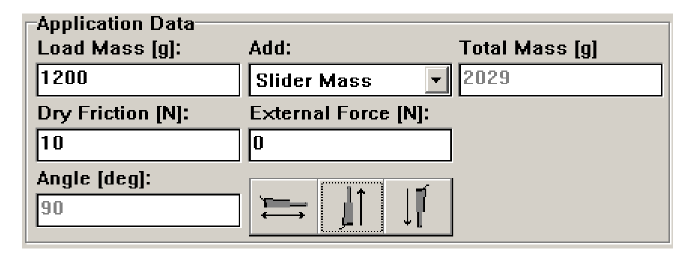


Abbildung 7-7: Eingabe der Applikationsdaten



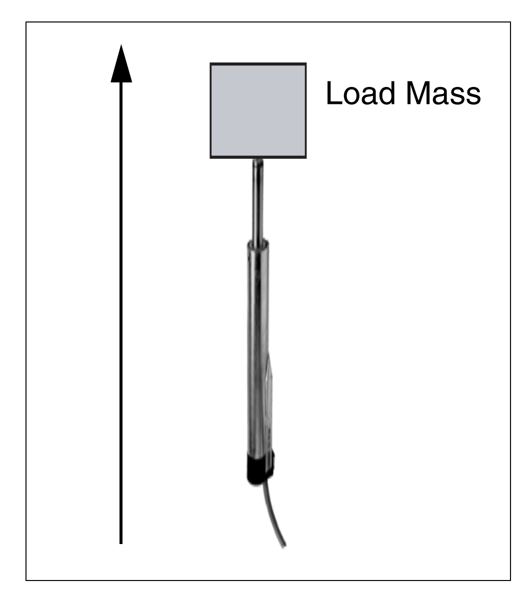


Abbildung 7-8: Positive Richtung des LinMot Antriebes

Berechnete Reglerparameterwerte

Die berechenten Werte für die Feed Forward Parameter unnd den Stromoffset werden in der Gruppe "Control Parameters" (siehe Abbildung 7-9, "Berechnete Reglerparameterwerte"). Um diese Werte auf einen Motor zu übernehmen klicken Sie auf die Taste "Apply to". Wenn Sie mit dem Servo-Controller verbunden sind werden die Werte direkt geschrieben und sind sofort aktiv (Live-Parameter).

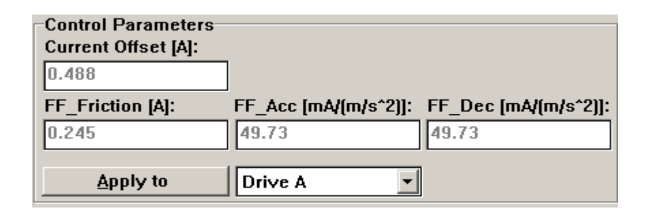


Abbildung 7-9: Berechnete Reglerparameterwerte

Tip:

Wenn die Masse einer Applikation nicht bekannt ist, kann folgendermasse vorgegangen werden: Den Motor an einer bestimmten Position stoppen und den gestellten Motorenstrom mit dem Oscilloskop auslesen. Oder den Maximalstrom reduzieren, bis der Motor die Position knapp nicht mehr halten kann.

7.8 Einstellen des max. Stromes (max Current)

Der maximale Strom sollte an den verwendeten Linearmotor angepasst werden und kann mit dem Parameter **Maximal Current** im Verzeichnis **\Drives\Drive X\Control Parameters** eingestellt werden. Dabei sollten die Werte gemäss der folgenden Tabelle eingesetzt werden:

	Serie E100		Serie E1000 /E1001	
Motortyp	**24V Speisung**	**48V Speisung**	**48V Speisung**	**72V Speisung**
P01-23x80/…	2.0A	3.0A	3.0 A	3.0 A
P01-23x160/…	1.0A	2.0A	2.0A	2.8A
P01-37x120/…	—	3.0A	6.0A	6.0A
P01-37x240/…	—	3.0A	3.3A	5.0A

Werden kleinere Werte eingestellt, reduziert sich die gemäss Datenbuch spezifizierte Spitzenkraft F_P der Motoren. Grössere Werte als die in der Tabelle aufgeführten, führen zu einem schlechteren Regelverhalten bei gleichbleibender Spitzenkraft. Zu beachten ist, dass bei der E100 Controller Serie im Verzeichnis **\Drives\Drive X\Control Switches** der Strombereich zudem auf 2 bzw. 3 A eingestellt werden muss.



7.9 Grundeinstellungen des Reglers

Die nachfolgenden Einstellungen genügen meist für einfache Anwendungen. Zu beachten ist, dass bei der E100 Servo Controller Serie im Verzeichnis **\Drives\Drive X\Control Switches** der Strombereich auf 2 A bzw. 3 A eingestellt werden muss.

	PS01 - 23x80	PS01- 23x160	PS01- 37x120	PS01- 37x240
Max Current	2.99 A	2 A	2.99 A	2.99 A
Current Offset	0 A	0 A	0 A	0 A
P	1 A/mm	1 A/mm	1 A/mm	1 A/mm
D	4 A/(m/s)	4 A/(m/s)	4 A/(m/s)	4 A/(m/s)
I	0 A/(mm*s)	0 A/(mm*s)	0 A/(mm*s)	0 A/(mm*s)
FF Friction	0 A	0 A	0 A	0 A
FF Acceleration	0 mA/(m/s^2)	0 mA/(m/s^2)	0 mA/(m/s^2)	0 mA/(m/s^2)
FF Deceleration	0 mA/(m/s^2)	0 mA/(m/s^2)	0 mA/(m/s^2)	0 mA/(m/s^2)
Filter max Speed	0.781 m/s	0.781 m/s	0.781 m/s	0.781 m/s
Filter max Accel.	30.5 m/s^2	30.5 m/s^2	30.5 m/s^2	30.5 m/s^2
Control Switches	3 A	3 A	3 A	3 A

Abbildung 7-10: Grundeinstellung für E100 Controller bei 48V Speisung

	PS01 - 23x80	PS01- 23x160	PS01- 37x120	PS01- 37x240
Max Current	4 A	2.8 A	6 A	5 A
Current Offset	0 A	0 A	0 A	0 A
P	1 A/mm	1 A/mm	1 A/mm	1 A/mm
D	4 A*s/mm	4 A*s/mm	4 A*s/mm	4 A*s/mm
I	0 A/(mm*s)	0 A/(mm*s)	0 A/(mm*s)	0 A/(mm*s)
FF Friction	0 A	0 A	0 A	0 A
FF Acceleration	0 mA/(m/s^2)	0 mA/(m/s^2)	0 mA/(m/s^2)	0 mA/(m/s^2)
FF Deceleration	0 mA/(m/s^2)	0 mA/(m/s^2)	0 mA/(m/s^2)	0 mA/(m/s^2)
Filter max Speed	0.781 m/s	0.781 m/s	0.781 m/s	0.781 m/s
Filter max Accel.	30.5 m/s^2	30.5 m/s^2	30.5 m/s^2	30.5 m/s^2

Abbildung 7-11: Grundeinstellung für E1000 / E1001 Controller bei 72V Speisung



7.10 Optimierung des Reglers

Es gibt viele unterschiedliche Möglichkeiten einen PID-Regler optimal einzustellen. Nachfolgend eine bewährte Methode aus der Praxis:

4 Einstellen des maximalen Stromes für den gewählten Motortyp gemäss Kapitel “Einstellen des max. Stromes (max Current)” auf Seite 114.

5 Einstellung des Vorfilters gemäss Kapitel “Einstellung des Vorfilters (Filter)” auf Seite 106. bzw. der Sollwertkurve gemäss Kapitel “Sollwertvorgabe mittels Kurven” auf Seite 107.

6 Einstellen der Vorsteuerungsparameter (feed forward) gemäss Kapitel “Einstellung der Vorsteuerung (Feed-Forward)” auf Seite 108. und des Strom-offsets gemäss Kapitel “Einstellung des Stromoffsets (Current-Offset)” auf Seite 109.

7 Nach diesen zwei Schritten müssen die folgenden PID-Parameter im Verzeichnis **\Drives\Drive X\Control Parameters** eingestellt werden:

P = 0.25 A/mm   D = 2.00 A(m/s)   I = 0.00 A/(ms)

Nun kann die gewünschte Sollkurve geladen und der Motor im Modus **Continuous Curve** gestartet werden bzw. die externe Steuerung kann die Sollwerte kontinuierlich vorgeben.

8 Nun den Parameter **D** schrittweise um 1 erhöhen bis der Motor zu schwingen beginnt (Geräuschentwicklung). Dann den D Wert auf 60% reduzieren.

9 Nun den Parameter **P** schrittweise um 0.25 erhöhen bis der Motor zu schwingen beginnt. Dann den P Wert auf 80% reduzieren.

10 Der Parameter **I** sollte nur eingestellt werden, falls die stationäre Regelabweichung (Positionsdifferenz zwischen Ist- und Sollwert im Stillstand) nach dem Einstellen von P- und D- Anteil noch zu gross ist.
Zum Einstellen des Parameters **I** den Wert schrittweise um 5 erhöhen bis die stationäre Regelabweichung minimiert ist und gleichzeitig kein Überschwingen beim Beschleunigen und Bremsen entsteht.

7.11 Erfolgskontrolle

Die richtige Einstellung des Reglers wird am besten mit dem in der *LinMot*® Talk Software integrierten Oszilloskop überprüft. Dabei interessiert vor allem der Vergleich zwischen der aktuellen Position (Actual Position) und der Sollposition (Demand Position).
Im weiteren lohnt es sich, den Iststrom (actual current) mit dem integrierten Oszilloskop zu kontrollieren: Verbleibt der Strom lange in den Begrenzungen, ist dies ein Zeichen dafür, dass der Motor überlastet ist und damit ein sauberes Regelverhalten nicht mehr gewährleistet ist.



8. *LinMot®*ASCII-Protokoll

8.1 Einführung

Das *LinMot®* ASCII-Protokoll bietet dem Anwender die Möglichkeit mit einem einfachen ASCII-Protokoll über eine RS232- oder RS485-Schnittstelle Befehle an den *LinMot®* Servo Controller zu senden. Damit können *LinMot®* Servo Controller nahtlos in Systeme integriert werden, die mit Hilfe dieses Schnittstellenstandards betrieben werden.

Die folgenden Abbildungen zeigen typische Systemumgebungen, in denen *LinMot®* Servo Controller über ein ASCII-Protokoll betrieben werden.

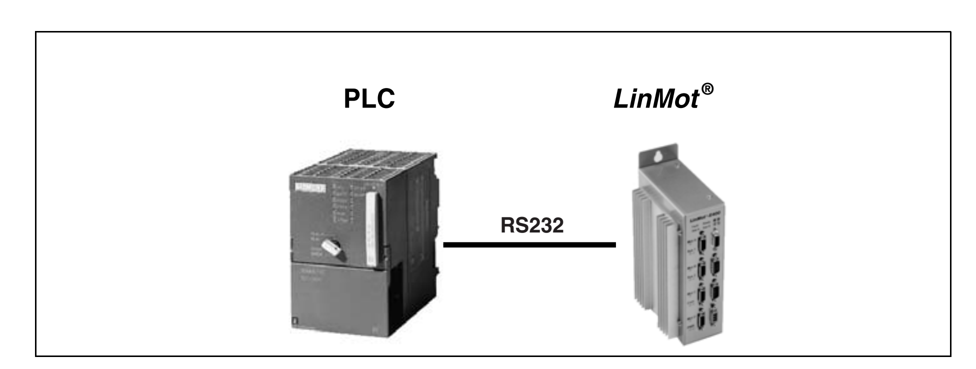


Abbildung 8-1: *LinMot®* Servo Controller mit RS232

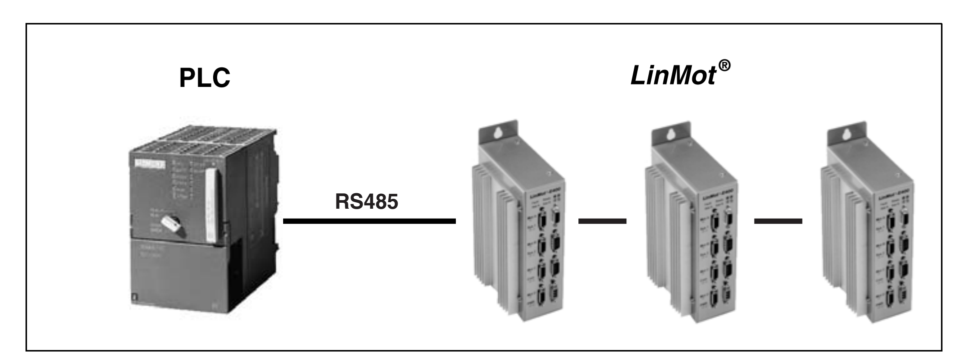


Abbildung 8-2: Mehrere mit RS485 vernetzte *LinMot®* Servo Controller

8.2 Projektierung und Installation

In diesem Kapitel wird die Projektierung und Installation von *LinMot®* Servo Controllern, welche mit dem ASCII-Protokoll betrieben werden sollen, erläutert.

Parametrierung

Um einen *LinMot®* Servo Controller über das ASCII-Protokoll anzusteuern, muss dieser entsprechend parametriert werden. Die Parametrierung geschieht mit Hilfe des Parameter-Inspectors der *LinMot®* Talk Software. Bei der Projektierung werden alle Parameter eingestellt, welche **nicht mit Hilfe des ASCII-Protokolles verändert** werden können:

- Befehls-Schnittstelle
- Motortyp
- Betriebsart
- Initialisierungsart
- Fehlerbehandlung

Sofern diese Parameter für die entprechende Anwendung richtig eingestellt worden sind und der Servo Controller korrekt mit dem PC oder der SPS verbunden ist, kann er über das ASCII-Protokoll betrieben werden.

Befehls-Schnittstelle

Dieser Parameter befindet sich im Parameterinspektor im Pfad **\System\Command Interface**. Er muss entweder auf **ASCII RS232** oder auf **ASCII RS485** eingestellt werden.

Motortyp

Über diese Parameter kann für jeden Motor festgelegt werden, um welchen Typ es sich handelt. Die Parameter finden sich im Verzeichnis **\Drives\Drive X\Type**. Das ‘X’ steht dabei für einen der maximal 4 möglichen Motoren ‘A’ bis ‘D’.

Initialisierungsart

Die Art der Initialisierung der Motoren wird durch die Parameter im Verzeichnis **\Drives\Drive X\Initialization** festgelegt.

RS232-Betrieb

Alle *LinMot®* Servo Controller werden ab Werk für den Betrieb mit der RS232-Schnittstelle konfiguriert. Die Belegung der Schnittstelle ist im Bedienerhandbuch beschrieben. Die RS232-Schnittstelle wird wie folgt betrieben:

Parameter	Wert
Baud Rate	9’600
Start Bits	1
Daten Bits	8
Stop Bits	1
Parity	keine

Beim RS232-Betrieb wird der Controller immer mit der Bus ID = 1 angesprochen, unabhängig davon, wie die Drehschalter ID0 und ID1 stehen.



RS485-Betrieb

Für den Betrieb über die RS485-Schnittstelle empfiehlt es einen neueren Controller zu verwenden, bei welchem sich die Drehcodierschalter auf der Unterseite und nicht auf der Rückseite befinden. Dies hat den Vorteil, dass die Geräte nicht geöffnet werden müssen, um die Jumperstellung zu ändern.

Falls trotzdem ein Controller mit rückseitigen Codierschaltern zum Einsatz kommen sollte, ist es empfehlenswert, mit dem *LinMot®* Support Kontakt aufzunehmen. (support@linmot.com)

Die RS485-Schnittstelle wird mit den folgenden Parametern betrieben:

Parameter	Wert
Baud Rate	9’600
Start Bits	1
Daten Bits	8
Stop Bits	1
Parity	keine

ID

Die ID des Servo Controllers kann mit Hilfe der Drehschalter auf der Unterseite eingestellt werden. Bei der Benutzung des ASCII-Protokolls via RS485 sind die Nummern 1 bis 6 für die ID0 (bei Version 3 Kontrollern ID LOW) zulässig. Die ID1 (ID HIGH) muss immer auf den Wert 0 eingestellt werden. Es können somit maximal 6 Servo Controller mit RS485 vernetzt werden. Bei der Benutzung der RS232 für das ASCII-Protokoll ist die Stellung der Drehschalter irrelevant.

Die folgende Abbildung zeigt die Unterseite des Controllers Auf den Version 3 Kontrollern befinden sich die Codierschalter auf der Frontseite.

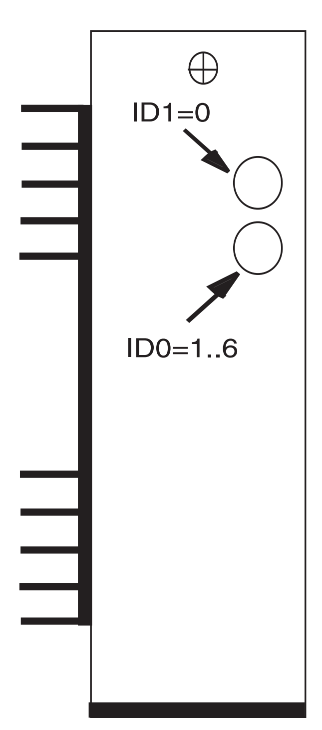


Abbildung 8-3: Einstellen der Servo Controller-ID für RS485-Betrieb



’Fail-save biasing’

Beim sog. ‘Fail-save biasing’ wird durch die Verwendung von Widerständen ein gesicherter Pegel auf dem Bus garantiert, auch wenn kein Treiber aktiv ist. Dies ist für die korrekte Funktionsweise notwendig. Die Beschaltung der Widerstände kann der unteren Abbildung entnommen werden.

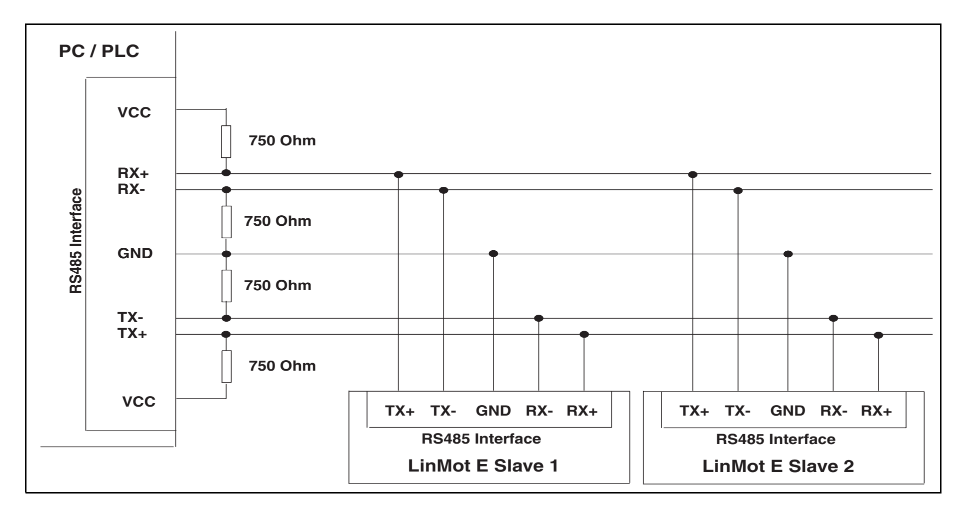


Abbildung 8-4: Verkabelung der RS485-Schnittstelle

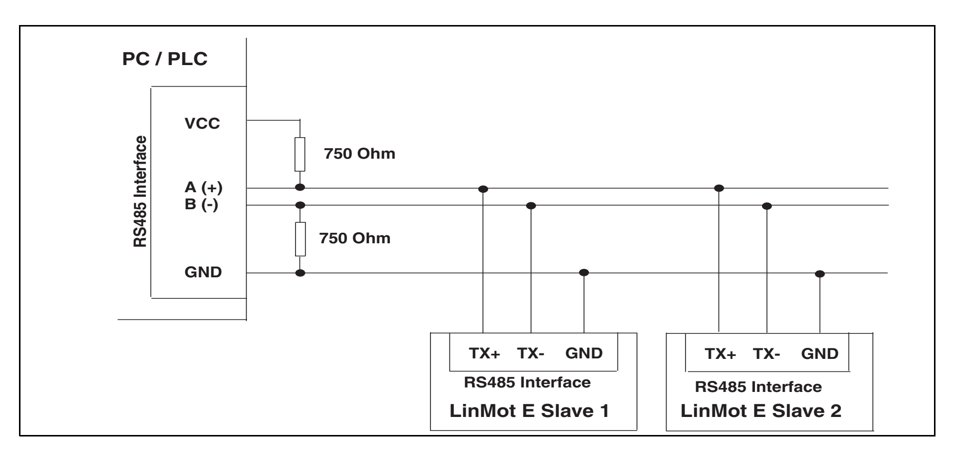


Abbildung 8-5: Verkabelung der RS485-Schnittstelle



Inbetriebnahme

Die folgenden Schritte zeigen, wie ein Servo Controller mit dem ASCII-Protokoll in Betrieb genommen wird:

1 Den Servo Controller gemäss Benutzerhandbuch Kapitel “RS232-Betrieb” auf Seite 118 oder Kapitel “RS485-Betrieb” auf Seite 119 konfigurieren. Wichtig ist dabei, dass die ID des Controllers richtig gesetzt wird. Damit die neue ID vom Controller erkannt wird, muss dieser neu aufgestartet werden (Power Off / Power On).

2 *LinMot®* Talk aufstarten und Servo Controller gemäss Kapitel “Parametrierung” auf Seite 118 parametrieren. Danach *LinMot®* Talk beenden.

3 Nun können dem Servo Controller ASCII-Komandos über die RS232 oder RS485-Schnittstelle gesendet werden (z.B. mit dem Hyperterminal Programm von Windows 95/NT).

Achtung

Die 'Backspace'- und 'Delete'-Taste darf in einem Terminalprogramm nicht benutzt werden um Kommandos an den Servo Controller zu senden!



8.3 Befehlsübersicht

ASCII Befehle				L	S	M	E
	Wert / Befehl	**Schreib Befehle**	**Lese Befehle**	*LinMot*	Stepper	Magnet	System
Schreib Befehle	Inkrementiere Sollposition	!IP	—	✘	✘		
	Inkrementiere Sollposition beim nächsten Trigger	!TI	—	✘	✘		
	Setze Sollposition beim nächsten Trigger	!TP	—	✘	✘	✘	
	Gehe zu Sollposition von aktueller Position	!SW	—	✘	✘	✘	
	Gehe zu Sollposition von aktueller Position beim nächsten Trigger	!TW	—	✘	✘	✘	
	Starte Kurve	!RC	—	✘	✘	✘	
	Starte Kurve auf Trigger	!TC	—	✘	✘	✘	
	Starte Kurve zyklisch	!CC	—	✘	✘	✘	
	Starte Kurve zyklisch auf Trigger	!CT	—	✘	✘	✘	
	Starte Kurve von aktueller Position	!RA	—	✘	✘	✘	
	Starte Kurve auf Trigger von aktueller Position	!TA	—	✘	✘	✘	
	Starte Kurve zyklisch von aktueller Position	!CA	—	✘	✘	✘	
	Starte Kurve zyklisch auf Trigger von aktueller Position	!CB	—	✘	✘	✘	
	Beende zyklische Kurve	!CS	—	✘	✘	✘	
	Verschiebe Referenzposition	!MH	—	✘	✘		
	Definiere neue Istposition	!RP	—	✘	✘		
	Setze Sollposition auf Istposition	!RQ	—	✘	✘		
	Setze internen Positionszähler auf den Wert 0	!ZD	—	✘	✘		
	Setze Adress Segment Offset	!AO	—				✘
	Setze Adress Segment Nummer	!AS	—				✘
	Schreibe Datenwort	!WR	—				✘
	Schreibe Datenwort mit Adressinkrementierung	!WS	—				✘
Lese/Schreib Befehle	Sollposition	!SP	!GD	✘	✘	✘	
	FF Acceleration	!DA	!EA	✘			
	FF Deceleration	!DB	!EB	✘			
	FF Friction	!DF	!EF	✘			
	P-Wert von Regler	!DP	!EP	✘			
	D-Wert von Regler	!DD	!ED	✘			
	I-Wert von Regler	!DI	!EI	✘			
	Maximale Geschwindigkeit	!SV	!GV	✘	✘		
	Maximale Beschleunigung	!SA	!GA	✘	✘		
	Maximaler Strom	!SC	!GC	✘	✘	✘	
	Strom Offset	!DK	!EK	✘			
	Kurven-Amplitude	!DC	!EC	✘	✘	✘	
	Kurven-Offset	!DO	!EO	✘	✘	✘	
	Kurven-Geschwindigkeit	!DS	!ES	✘	✘	✘	
	Flag **FREEZE**	!SF	!GX	✘	✘	✘	✘
	Flag **INIT** / Flag **RUN** / Flag **STOP**	!SI / !SR / !SS	!GX				✘



ASCII Befehle

				L	S	M	E
Lese Befehle	Iststrom	—	!AC	✘			
	Istposition	—	!GP	✘	✘	✘	
	Positionsauflösung	—	!PI	✘	✘	✘	
	Geschwindigkeitauflösung	—	!VI	✘	✘		
	Beschleunigungsauflösung	—	!AI	✘	✘		
	Stromauflösung	—	!CI				✘
	Status	—	!GS				✘
	System Fehler Status	—	!GE				✘
	System Warn Status	—	!GW				✘
	Motor Fehler Status	—	!EE	✘	✘	✘	
	Motor Warn Status	—	!EW	✘	✘	✘	
	Zustand-Flags	—	!EX				✘
	Protokollversion	—	!PV				✘
	Lese Datenwort	—	!RD				✘
	Lese Datenwort mit Adressinkrementierung	—	!RE				✘

8.4 Befehlsaufbau

Alle ASCII-Befehle sind nach dem unten abgebildeten Muster aufgebaut:

Aufbau Befehl		
Byte	**Wert**	**Bedeutung**
0	'!'	Befehlskopf
1…2	char, char	Befehl
3…x	[char], …	Argumente
x+1	'↵' (0x0D)	Befehlsabschluss ASCII Zeichen 0x0D (= 13 dezimal) Carriage Return

Jeder Befehl fängt mit einem Ausrufezeichen an, danach kommen zwei Zeichen, welche den Befehl kodieren, anschliessend folgen die optionalen Befehlsargumente und als Abschluss steht ein Zeilenende-Zeichen.

Jeder auf dem *LinMot*® Servo Controller empfangene Befehl wird quittiert. Ein weiterer Befehl darf nur gesendet werden, falls der letzte Befehl vom Servo Controller quittiert wurde.

Die Quittierung eines Befehls ist dabei nach dem folgenden Muster aufgebaut:

Aufbau Quittierung		
Byte	**Wert**	**Bedeutung**
0	'#'	Quittierungskopf
1…x	char, …	Quittierungsmeldung
x+1	'↵' (0x0D)	Quittierungsabschluss

Das folgende Beispiel zeigt, wie eine neue Sollposition für den Motor A dem angeschlossenen Servo Controller vorgegeben wird.

Beispiel		
Richtung	**ASCII-Sequenz**	**Beschreibung**
PC -> *LinMot*® Controller	'!SP2000A' + 0x0D	Setze die Sollposition von Motor A auf 2000 Inkremente.
LinMot® Controller -> PC	'#' + 0x0D	Wenn das '#'-Zeichen gefolgt von einem '↵'-Zeichen (0x0D) übermittelt wird, bedeutet dies, dass der Befehl vom *LinMot*® Controller aktzeptiert wurde.

Hinweis

Es können über RS232 das ASCII- und das LinMot-Talk-Protokoll miteinander gefahren werden. Wichtig ist aber, dass die Befehle mit Antwort ganz abgeschlossen sind, bevor ein Befehl aus dem anderen Protokoll gesendet wird.

Falls bei der Kommunikation mit dem LinMot-Talk-Protokoll der Fehler "Device TimeOut" auftreten sollte, wurde höchstwahrscheinlich ein ASCII-Befehl nicht abgeschlossen. Dann muss einfach mit dem Terminalprogramm ein Carriage Return (0x0D) gesendet werden.



8.5 Befehle

Die folgende Tabelle gibt eine Übersicht über die in den folgenden Kapiteln verwendeten Variablentypen. Die Variablentypen werden tiefgestellt und in eckige Klammern gesetzt hinter der Variable angefügt. So steht z.B. $pos_{[sint16]}$ für eine Variable mit dem Namen *pos* und dem Variablen-Typ von *sint16*. Der Variablen-Typ *sint16* steht, wie aus der untenstehenden Tabelle entnommen werden kann, für einen vorzeichenbehafteten Kardinalwert. In eckige Klammern gesetzte Werte sind optional. Das '|'-Zeichen steht für ein 'oder'.

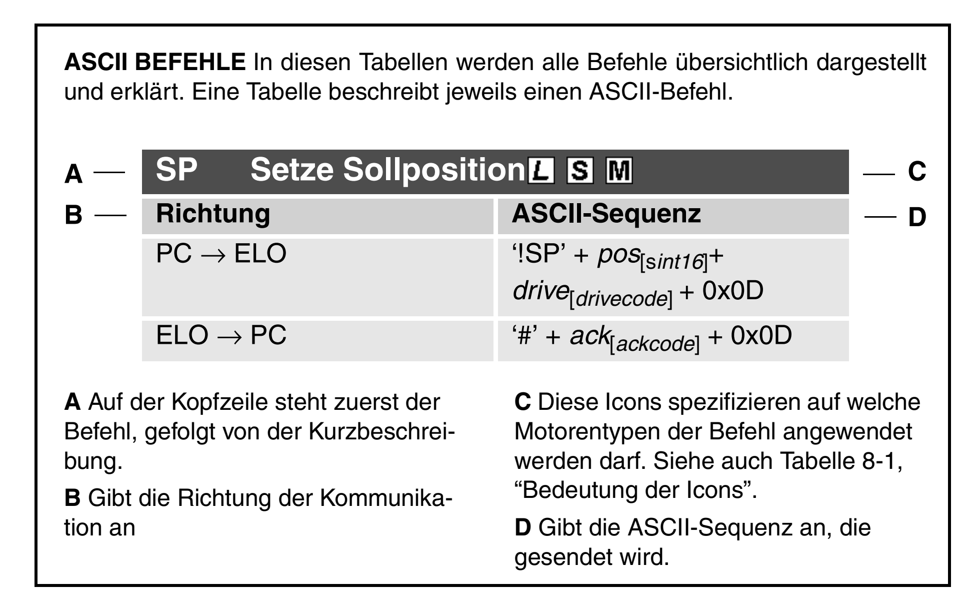
Argumenttypen

Argumenttypen	
Argumenttyp	**Beschreibung**
uint16	Kardinalwert mit dem Bereich 0 … 2^{16}-1. Der Wert muss dezimal im ASCII-System repräsentiert werden. Das Plus-Zeichen bei positiven Werten ist optional. Bsp.: 0, 123, 3300, +200, 500
sint16	Vorzeichenbehafteter Kardinalwert mit dem Bereich -2^{15} … 2^{15}-1. Der Wert muss dezimal im ASCII-System repräsentiert werden. Das Plus-Zeichen bei positiven Werten ist optional. Bsp.: 0, 123, -2000, -200, +240
uint32	Kardinalwert mit dem Bereich 0 … 2^{32}-1. Der Wert muss dezimal im ASCII-System repräsentiert werden. Bsp.: 0, 123, 200000, 3000000, +240
ackcode	Quittierungscode, der vom Controller nach jedem empfangenen Aktions- oder Set-Befehl zurückgeschickt wird. Die einzelnen Codes werden in der Referenztabelle in Kapitel "Referenztabelle: Status- und Fehlermeldungen" auf Seite 152 erklärt.
drivecode	Motorbezeichner. Motorbezeichner sind ASCII-Grossbuchstaben. Wenn das Protokoll auf RS232 eingestellt ist, sind nur die Motorenbezeichner 'A', 'B', 'C' und 'D' erlaubt. Ansonsten können, falls der entsprechende Controller und der Motor existieren, beliebige Bezeichner verwendet werden (von 'A' bis 'X'). Alle Zuordnungen befinden sich im Kapitel "Referenztabelle: Motorbezeichner" auf Seite 154. Bsp.: A … D => Drive A, B, C, D (Controller mit ID = 1) E … H => Drive A, B, C, D (Controller mit ID = 2) … U … X => Drive A, B, C, D (Controller mit ID = 6)
elocode	Controllerbezeichner. Controllerbezeichner sind ASCII-Ziffern. Wenn das Protokoll auf RS232 eingestellt ist, ist nur die Ziffer '1' erlaubt. Ansonsten können, falls die entsprechenden Controller existieren, die Ziffern '1' bis '6' verwendet werden.
statecode	Statuscode, welcher vom Servo Controller nach einem 'GS'-Befehl zurückgesendet wird. Die einzelnen Codes werden in Kapitel "Referenztabelle: Status- und Fehlermeldungen" auf Seite 152 erklärt.



Befehlsbeschreibung

Alle Befehle in diesem Kapitel sind alphabetisch geordnet. Die folgende Beschreibung zeigt nach welchem Muster die Befehlsbeschreibung aufgebaut ist.



Die folgende Tabelle zeigt welche Bedeutung die Icons in der Kopfzeile der Befehlsbeschreibung haben.

Icon	Bedeutung
E	Der Befehl gilt für den Kontroller
L	Der Befehl gilt für Linearmotoren
S	Der Befehl gilt für Schrittmotoren
M	Der Befehl gilt für Magnete / Spulen

Tabelle 8-1: Bedeutung der Icons



AC

AC Lese Sollstrom	L
Richtung	**ASCII-Sequenz**
PC → ELO	'!AC' + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $curr_{[sint16]}$ + 0x0D

Dieser Befehl erfragt den vom Regler eingestellten Stromwert. Die Auflösung des Stromes kann mit dem Befehl **CI** gelesen werden.

Wert	Min	Max
$curr_{[sint16]}$	-256	256

AI

AI Lese Beschleunigungsauflösung	E S
Richtung	**ASCII-Sequenz**
PC → ELO	'!AI' + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $accres_{[uint32]}$ + 0x0D

Dieser Befehl erfragt die Beschleunigungsauflösung. Er darf nur bei Linear- und Schrittmotoren verwendet werden. Die Werte werden dabei, je nach gewähltem Motortyp, in verschiedenen Einheiten zurückgegeben:

Motortyp	Einheit
LinMot®	1µm/s²
Schrittmotor	2^{-10} Schritte/s²

Beispielsequenz:

PC → ELO	ELO → PC	Beschreibung
!AIB↵	#238419↵	Erfragt aktuelle Beschleunigungsauflösung von Motor B. Der zurückgegebene Wert entspricht 0.238m/s², da der 'Motor B' in diesem Beispiel als *LinMot®* konfiguriert ist.

AO

AO Setze Adress Segment Offset	E
Richtung	**ASCII Sequenz**
PC → ELO	'!AO' + $segoffset_{[uint16]}$+ $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt den Adress Segment Offset für Schreib- oder Lesebefehle. Die komplette Adresse ist 24 Bit breit und besteht aus der Segment Nummer (oberste 8 Bit) und dem Segment Offset (unterste 16 Bit). Der Segment Offset muss eine gerade Zahl sein. Die drive-Bezeichnung wird zur Adressierung des Controller verwendet.

Wert	Min	Max
$segoffset_{[uint16]}$	0	65534

AS

AS Setze Adress Segment Nummer	E
Richtung	**ASCII Sequenz**
PC → ELO	‘!AS’ + $seg_{[uint8]}$+ $drive_{[drivecode]}$ + 0x0D
ELO → PC	‘#’ + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt die Adress Segment Nummer für Schreib- oder Lesebefehle. Die komplette Adresse ist 24 Bit breit und besteht aus der Segment Nummer (oberste 8 Bit) und dem Segment Offset (unterste 16 Bit). Die drive-Bezeichnung wird zur Adressierung des Controller verwendet.

Wert	Min	Max
$seg_{[uint8]}$	0	255

CA

CA Starte Kurve zyklisch von aktueller Position	L S M
Richtung	**ASCII-Sequenz**
PC → ELO	‘!CA’ + $curve_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	‘#’ + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl startet eine Kurve zyklisch von der aktuellen Sollposition auf. Das bedeutet, dass die Kurve nach Beendigung sofort wieder aufgestartet wird. Mit dem Befehl **CS** kann die Ausführung der Kurve beendet werden. Solange die Kurve zyklisch aufgestartet wird dürfen keine anderen Bewegungsbefehle für diesen Motor benutzt werden. Dieser Befehl kann nur im Zustand **RUN** benutzt werden.

Achtung: Dieser Befehl ändert den Parameter Curve Position Offset.

CB

CB Starte Kurve zyklisch beim nächsten Trigger von aktueller Position	L S M
Richtung	**ASCII-Sequenz**
PC → ELO	‘!CB’ + $curve_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	‘#’ + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl startet bei der nächsten positiven Flanke des Triggersignales eine Kurve zyklisch von der aktuellen Sollposition auf. Nach dem ersten Aufstarten wird die Kurve zyklisch abgefahren ohne dass das Triggersignal beachtet wird. Mit dem Befehl CS kann die zyklische Ausführung der Kurve beendet werden. Solange die Kurve zyklisch aufgestartet wird dürfen keine anderen Bewegungsbefehle für diesen Motor benutzt werden.

Achtung: Dieser Befehl ändert den Parameter Curve Position Offset.



CC

CC Starte Kurve zyklisch	L S M
Richtung	**ASCII-Sequenz**
PC → ELO	'!CC' + $curve_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl startet eine Kurve zyklisch auf. Das bedeutet, dass die Kurve nach Beendigung sofort wieder aufgestartet wird. Mit dem Befehl **CS** kann die Ausführung der Kurve beendet werden. Solange die Kurve zyklisch aufgestartet wird dürfen keine anderen Bewegungsbefehle für diesen Motor benutzt werden. Dieser Befehl kann nur im Zustand **RUN** benutzt werden.

CI

CI Lese Stromauflösung	E
Richtung	**ASCII-Sequenz**
PC → ELO	'!CI' + $elo_{[elocode]}$ + 0x0D
ELO → PC	'#' + $currres_{[uint32]}$ + 0x0D

Dieser Befehl erfragt die Stromauflösung des gewählten Servo Controllers. Die Werte werden in µA zurückgegeben.

Beispielsequenz:

PC → ELO	ELO → PC	Beschreibung
!CI1↵	#23438↵	Erfragt Stromauflösung des Servo Controllers mit ID 1. Der zurückgegebene Wert entspricht 23.438mA.

CS

CS Beende zyklische Kurve	L S M
Richtung	**ASCII-Sequenz**
PC → ELO	'!CS' + $stop_{[drivesel]}$ + $anydrive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl beendet die Ausführung einer oder mehrerer zyklischer Kurven. Die Kurven laufen nach dem Empfang des Befehls noch bis zu ihrem Ende und werden dann nicht mehr erneut aufgestartet. In der Variablen *drivesel* kann spezifiziert werden, welche Motoren gestoppt werden sollen:

Bit	3	2	1	0
Name	Drive D	Drive C	Drive B	Drive A

Mit der Variablen *anydrive* wird der Controller ausgewählt, für welchen der Befehl gilt. Aus syntaktischen Gründen muss dafür ein beliebiger konfigurierter Motor dieses Controllers angegeben werden.

Beispielsequenz:

PC → ELO	ELO → PC	Beschreibung
!CS3A↵	#↵	Beendet die Ausführung der zyklischen Kurven auf den Motoren A und B.

CT

CT Starte Kurve zyklisch beim nächsten Trigger L S M	
Richtung	**ASCII-Sequenz**
PC → ELO	'!CT' + $curve_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl startet bei der nächsten positiven Flanke des Triggersignales eine Kurve zyklisch auf. Nach dem ersten Aufstarten wird die Kurve zyklisch abgefahren ohne dass das Triggersignal beachtet wird. Mit dem Befehl CS kann die zyklische Ausführung der Kurve beendet werden. Solange die Kurve zyklisch aufgestartet wird dürfen keine anderen Bewegungsbefehle für diesen Motor benutzt werden.

DA

DA Setze FF Acceleration-Wert des Reglers L	
Richtung	**ASCII-Sequenz**
PC → ELO	'!DA' + $ffacc_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt den **FF Acceleration**-Wert des Motors *drive* auf den Wert ff*acc*. Die Einheit beträgt 0.1 mA/(m/s^2).

Wert	**Min**	**Max**
$ffacc_{[uint16]}$	0	32640

DB

DB Setze FF Deceleration-Wert des Reglers L	
Richtung	**ASCII-Sequenz**
PC → ELO	'!DB' + $ffdec_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt den **FF Deceleration**-Wert des Motors *drive* auf den Wert *ffdec*. Die Einheit beträgt 0.1 mA/(m/s^2).

Wert	**Min**	**Max**
$ffdec_{[uint16]}$	0	32640



DC

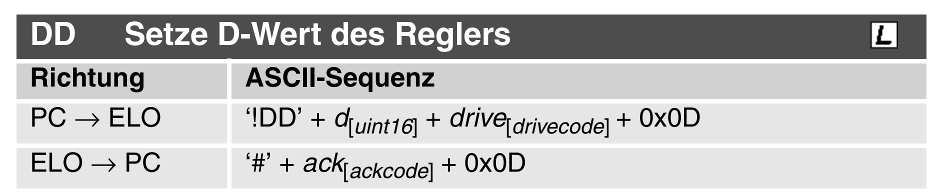
DC	Setze Kurven-Amplitude [L] [S] [M]
Richtung	**ASCII-Sequenz**
PC → ELO	‘!DC’ + $camp_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	‘#’ + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt die Kurven-Amplitude des gewählten Motors. Dabei entspricht der maximale Wert (4096) dem Skalierungsfaktor 100%. Bei diesem Faktor ist daher die Amplitude der Kurve so gross, wie sie im Kurveneditor erstellt wurde. Kleinere Werte verkleinern die Kurve linear.

Achtung: Wenn dieser Befehl verwendet wird während eine Kurve läuft, wird der Motor springen.

Wert	Min	Max
$camp_{[uint16]}$	0	4096

DD



DD	Setze D-Wert des Reglers [L]
Richtung	**ASCII-Sequenz**
PC → ELO	‘!DD’ + $d_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	‘#’ + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt den **D**-Wert des Motors *drive* auf den Wert *dec*. Die Einheit beträgt 0.015A*s/mm.

Wert	Min	Max
$d_{[uint16]}$	0	32640

DF

DF	Setze FF Friction-Wert des Reglers [L]
Richtung	**ASCII-Sequenz**
PC → ELO	‘!DF’ + $fffri_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	‘#’ + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt den **FF Friction**-Wert des Motors *drive* auf den Wert *fffri*. Die Einheit beträgt 23.4mA.

Wert	Min	Max
$fffri_{[uint16]}$	0	255

DI

DI Setze I-Anteil des Reglers	L
Richtung	**ASCII-Sequenz**
PC → ELO	'!DI' + $i_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt den **I**-Wert des Motors *drive* auf den Wert *i*. Die Einheit beträgt 0.0457 A/(mm*s).

Wert	Min	Max
$i_{[uint16]}$	0	32640

DK

DK Setze Stromoffset	L
Richtung	**ASCII-Sequenz**
PC → ELO	'!DK' + $curroff_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt den Stromoffset des Motors *drive* auf den Wert *curroff*. Die Auflösung des Stroms kann mit dem 'CI'-Befehl erfragt werden. Ein Stromoffset ist sinnvoll, wenn der Motor bei einer vertikalen Einbaulage mit einer Lastmasse belastet wird.

Wert	Min	Max
$curroff_{[uint16]}$	-256	256

DO

DO Setze Kurven-Offset	L S M
Richtung	**ASCII-Sequenz**
PC → ELO	'!DO' + $coff_{[sint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt den Kurven-Offset des gewählten Motors. Die Auflösung der Position kann mit dem Befehl **PI** abgefragt werden.

Wert	Min	Max
$coff_{[sint16]}$	-32256	+32256

DP

DP Setze P-Anteil des Reglers	L
Richtung	**ASCII-Sequenz**
PC → ELO	'!DP' + $p_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt den **P**-Wert des Motors *drive* auf den Wert *p*. Die Einheit beträgt 2.34 mA/mm.

Wert	Min	Max
$p_{[uint16]}$	0	32640



DS

DS	Setze Geschwindigkeit von Kurven	L S M
Richtung	**ASCII-Sequenz**	
PC → ELO	'!DS' + $vel_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D	
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D	

Dieser Befehl setzt die Geschwindigkeit von Kurven. Wenn der maximale Wert gesetzt ist, wird die Kurve so schnell abgefahren wie sie erstellt wurde. Bei kleineren Werten geht die Geschwindigkeit linear zurück. Die Geschwindigkeit kann zu einem beliebigen Zeitpunkt verändert werden.

Wert	Min	Max
$vel_{[uint16]}$	0	4096

EA

EA	Lese FF Acceleration-Wert des Reglers	L
Richtung	**ASCII-Sequenz**	
PC → ELO	'!EA'+ $drive_{[drivecode]}$ + 0x0D	
ELO → PC	'#' + $ffacc_{[uint16]}$ + 0x0D	

Dieser Befehl gibt den **FF Acceleration**-Wert des Motors *drive* zurück. Die Einheit beträgt 0.1 mA/(m/s^2).

Wert	Min	Max
$ffacc_{[uint16]}$	0	32640

EB

EB	Lese FF Deceleration-Wert des Reglers	L
Richtung	**ASCII-Sequenz**	
PC → ELO	'!EB' + $drive_{[drivecode]}$ + 0x0D	
ELO → PC	'#' + $ffdec_{[uint16]}$ + 0x0D	

Dieser Befehl gibt den **FF Deceleration**-Wert des Motors *drive* zurück. Die Einheit beträgt 0.1 mA/(m/s^2).

Wert	Min	Max
$ffdec_{[uint16]}$	0	32640

EC

EC	Lese Kurven-Amplitude	L S M
Richtung	**ASCII-Sequenz**	
PC → ELO	'!EC' + $drive_{[drivecode]}$ + 0x0D	
ELO → PC	'#' + $camp_{[uint16]}$ + 0x0D	

Dieser Befehl liest Kurven-Amplitude des gewählten Motors. Dabei entspricht der maximale Wert (4096) dem Skalierungsfaktor 100%.

Wert	Min	Max
$camp_{[uint16]}$	0	4096

ED

ED Lese D-Anteil des Reglers	**L**
Richtung	**ASCII-Sequenz**
PC → ELO	'!ED' + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $d_{[uint16]}$ + 0x0D

Dieser Befehl gibt den **D**-Wert des Motors *drive* zurück. Die Einheit beträgt 0.015A*s/mm.

Wert	Min	Max
$d_{[uint16]}$	0	32640

EE

EE Lese Motor Fehler Status	**L S M**
Richtung	**ASCII-Sequenz**
PC → ELO	'!EE' + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $state_{[statecode]}$ + 0x0D

Dieser Befehl liest den aktuellen **Motor Fehler Status** des gewählten Motors. Dabei haben die einzelnen Bits von *state* die folgende Bedeutung:

Bit	7	6	5	4	3	2	1	0
Name	Kein / Inkompatibles Bewegungsprofil	Falscher Motortyp	Initialisierung misslungen	reserviert	Läufer fehlt	Schlepp fehler	Motor heiss Sensor	Motor heiss berechnet

EF

EF Lese FF Friction-Wert des Reglers	**L**
Richtung	**ASCII-Sequenz**
PC → ELO	'!EF' + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $fffri_{[uint16]}$ + 0x0D

Dieser Befehl gibt den **FF Friction**-Wert des Motors *drive* zurück. Die Auflösung der Position kann mit dem Befehl **CI** abgefragt werden.

Wert	Min	Max
$fffri_{[uint16]}$	0	255



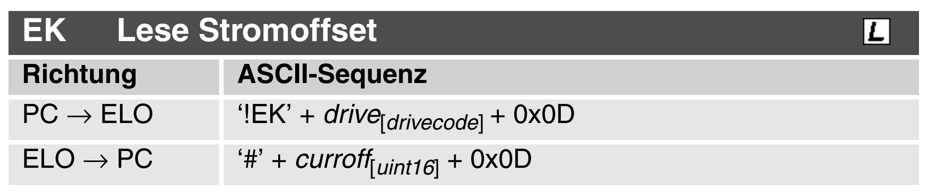
EI

EI	Lese I-Anteil des Reglers [L]
Richtung	**ASCII-Sequenz**
PC → ELO	'!EI' + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + i$_{[uint16]}$ + 0x0D

Dieser Befehl gibt den **I**-Wert des Motors *drive* zurück. Die Einheit beträgt 45.7 mA/(mm*s).

Wert	Min	Max
$i_{[uint16]}$	0	32640

EK



EK	Lese Stromoffset [L]
Richtung	**ASCII-Sequenz**
PC → ELO	'!EK' + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *curroff*$_{[uint16]}$ + 0x0D

Dieser Befehl liest den Stromoffset des Motors *drive* zurück. Die Auflösung des Stroms kann mit dem Befehl **CI** erfragt werden.

Die folgende Tabelle gibt an welche Minimal- bzw. Maximalwerte zugelassen sind.

Wert	Min	Max
$curroff_{[uint16]}$	-256	256

EO

EO	Lese Kurven-Offset [L] [S] [M]
Richtung	**ASCII-Sequenz**
PC → ELO	'!EO' + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *coff*$_{[sint16]}$ + 0x0D

Dieser Befehl liest den Kurven-Offset des gewählten Motors. Die Auflösung der Position kann mit dem Befehl **PI** abgefragt werden.

Wert	Min	Max
$coff_{[sint16]}$	-32256	+32256

EP

EP	Lese P-Anteil des Reglers [L]
Richtung	**ASCII-Sequenz**
PC → ELO	'!EP' + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *p*$_{[uint16]}$ + 0x0D

Dieser Befehl gibt den **P**-Wert des Motors *drive* zurück. Die Einheit beträgt 2.34 mA/mm.

Wert	Min	Max
$p_{[uint16]}$	0	32640

ES

ES Lese Geschwindigkeit von Kurven	L S M
Richtung	**ASCII-Sequenz**
PC → ELO	'!ES' + *drive*[*drivecode*] + 0x0D
ELO → PC	'#' + *vel*[*uint16*] + 0x0D

Dieser Befehl gibt die Geschwindigkeit von Kurven zurück. Wenn der maximale Wert gesetzt ist, wird die Kurve so schnell abgefahren wie sie erstellt wurde. Bei kleineren Werten geht die Geschwindigkeit linear zurück.

Wert	Min	Max
vel[*uint16*]	0	4096

EW

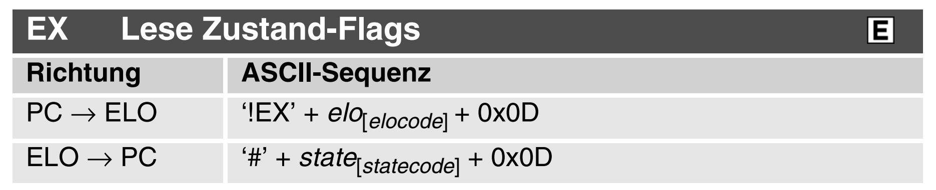
EW Lese Motor Warn Status	L S M
Richtung	**ASCII-Sequenz**
PC → ELO	'!EW' + *drive*[*drivecode*] + 0x0D
ELO → PC	'#' + *state*[*statecode*] + 0x0D

Dieser Befehl liest den aktuellen **Motor Warn Status** des gewählten Motors. Dabei haben die einzelnen Bits von *state* die folgende Bedeutung:

Bit	8	7	6	5	4	3	2	1	0
Name	Motor nicht im definierten Positionsbereich	reserviert	Initialisierung noch nicht ausgeführt	reserviert	reserviert	Läufer fehlt	Schleppfehler	Motor heiss Sensor	Motor heiss berechnet

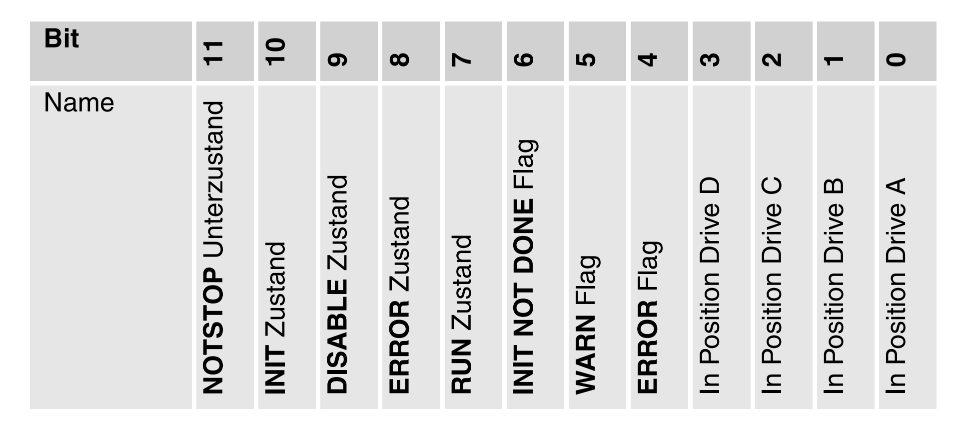


EX



EX Lese Zustand-Flags	E
Richtung	**ASCII-Sequenz**
PC → ELO	'!EX' + $elo_{[elocode]}$ + 0x0D
ELO → PC	'#' + $state_{[statecode]}$ + 0x0D

Dieser Befehl liest die Zustand-Flags des gewählten Servo Controllers. Dabei haben die einzelnen Bits von *state* die folgende Bedeutung:



Bit	Name
11	**NOTSTOP** Unterzustand
10	**INIT** Zustand
9	**DISABLE** Zustand
8	**ERROR** Zustand
7	**RUN** Zustand
6	**INIT NOT DONE** Flag
5	**WARN** Flag
4	**ERROR** Flag
3	In Position Drive D
2	In Position Drive C
1	In Position Drive B
0	In Position Drive A

Beispielsequenz:

PC → ELO	ELO → PC	Beschreibung
!EX1↵	#129↵	Der zurückgelesene Wert zeigt an, dass sich der Servo Controller im Zustand **RUN** befindet und der Motor A in Position ist.

GA

GA Lese maximale Beschleunigung	L S
Richtung	**ASCII-Sequenz**
PC → ELO	'!GA' + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $acc_{[uint16]}$ + 0x0D

Dieser Befehl gibt die maximale Beschleunigung des Motors *drive* zurück. Die Auflösung der Beschleunigung kann mit dem 'AI'-Befehl erfragt werden.

Wert	Min	Max
$acc_{[uint16]}$	1	1536

GC

GC Lese maximalen Strom (Kraft) L S M

Richtung	ASCII-Sequenz
PC → ELO	'!GC' + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $curr_{[uint16]}$ + 0x0D

Dieser Befehl gibt den maximalen Strom des Motors *drive* zurück. Die Auflösung des Stroms kann mit dem 'CI'-Befehl erfragt werden.

Die folgende Tabelle gibt an welche Minimal- bzw. Maximalwerte zugelassen sind.

Wert	Min	Max
$curr_{[uint16]}$	0	256

GD

GD Lese Sollposition L S M

Richtung	ASCII-Sequenz
PC → ELO	'!GD' + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $pos_{[sint16]}$ + 0x0D

Dieser Befehl gibt die Sollposition für den Motor *drive* zurück. Dieser Befehl kann nicht gebraucht werden, wenn ein Bewegungsprofil abgefahren wird. Die Auflösung der Position kann mit dem Befehl **PI** abgefragt werden.

Wert	Min	Max
$pos_{[sint16]}$	-32256	32256

GE

GE Lese System Fehler Status E

Richtung	ASCII-Sequenz
PC → ELO	'!GE' + $elo_{[elocode]}$ + 0x0D
ELO → PC	'#' + $state_{[statecode]}$ + 0x0D

Dieser Befehl liest den **System Fehler Status** des gewählten Servo Controllers. Dabei haben die einzelnen Bits von *state* die folgende Bedeutung:

Bit	7	6	5	4	3	2	1	0
Name	reserviert	reserviert	Elektronik Fehler[1]	DCLV Signal[2] zu hoch	DCLV Signal[2] zu tief	DCLV Power[3] zu hoch	DCLV Power[3] zu tief	reserviert

1) Der Kühlkörper des Controllers ist zu heiss (über 70° Celsius) oder ein Kurzschluss bei einer Motorphase wurde detektiert.
2) **DCLV Signal** steht für die Zwischenkreisspannung des Signalteils.
3) **DCLV Power** steht für die Zwischenkreisspannung des Leistungsteils



GP

GP Lese Istposition	L S M
Richtung	**ASCII-Sequenz**
PC → ELO	'!GP' + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *pos*$_{[sint16]}$ + 0x0D

Dieser Befehl erfragt die Istposition des Motors *drive*. Die Auflösung der Position kann mit dem Befehl **PI** erfragt werden.

Wert	Min	Max
pos$_{[sint16]}$	-32256	32256

Beispielsequenz:

PC → ELO	ELO → PC	Beschreibung
!GPA↵	#256↵	Erfragt die aktuelle Position von Motor A.

GS

GS Lese Status	E
Richtung	**ASCII-Sequenz**
PC → ELO	'!GS' + *elo*$_{[elocode]}$ + 0x0D
ELO → PC	'#' + *state*$_{[statecode]}$ + 0x0D

Dieser Befehl erfragt den aktuellen Status des Controllers. Der Status setzt sich zusammen aus einem Buchstaben, der den Zustand kodiert, sowie einer Zahl. Die Zahl wird nur im Fehlerzustand übermittelt und schlüsselt den aktuellen Fehler auf. Die Zustandskodierung wird in der Referenztabelle im Kapitel "Referenztabelle: Status- und Fehlermeldungen" auf Seite 152 erklärt.

Beispielsequenz:

PC → ELO	ELO → PC	Beschreibung
!GS1↵	#R↵	Erfragt den aktuellen Status. Das 'R' bedeutet, dass sich der Controller im Zustand **RUN** befindet.

GV

GV Lese maximale Geschwindigkeit	L S
Richtung	**ASCII-Sequenz**
PC → ELO	'!GV' + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *vel*$_{[uint16]}$ + 0x0D

Dieser Befehl gibt die maximale Geschwindigkeit des Motors *drive* zurück. Die Auflösung der Geschwindigkeit kann mit dem Befehl **VI** erfragt werden.

Wert	Min	Max
vel$_{[uint16]}$	6	24576



GW

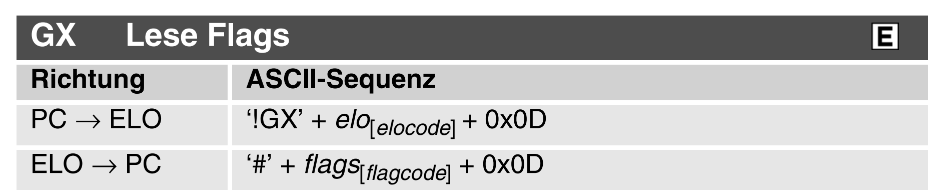
GW Lese System Warn Status	E
Richtung	**ASCII-Sequenz**
PC → ELO	'!GW' + *elo*$_{[elocode]}$ + 0x0D
ELO → PC	'#' + *state*$_{[statecode]}$ + 0x0D

Dieser Befehl liest den **System Warn Status** des gewählten Controllers. Dabei haben die einzelnen Bits von *state* die folgende Bedeutung:

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	reserviert	reserviert	Electronik Warnung[1]	DCLV Signal[2] hoch	DCLV Signal[2] tief	DCLV Power[3] hoch	DCLV Power[3] tief	reserviert

1) Der Kühlkörper des Servo Controllers ist zu heiss (über 70° Celsius) oder ein Kurzschluss bei einer Motorphase wurde detektiert. Nach fünf Sekunden geht der Servo Controller in den Zustand **ERROR**.
2) **DCLV Signal** steht für die Zwischenkreisspannung des Signalteils.
3) **DCLV Power** steht für die Zwischenkreisspannung des Leistungsteils

GX



GX Lese Flags	E
Richtung	**ASCII-Sequenz**
PC → ELO	'!GX' + *elo*$_{[elocode]}$ + 0x0D
ELO → PC	'#' + *flags*$_{[flagcode]}$ + 0x0D

Dieser Befehl liest alle Flags des gewählten Servo Controllers. Dabei haben die einzelnen Bits von *flags* die folgende Bedeutung:

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	**RUN** Flag	**INIT** Flag	**FREEZE** Flag (global)	**STOP** Flag	**FREEZE** Flag Drive D	**FREEZE** Flag Drive C	**FREEZE** Flag Drive B	**FREEZE** Flag Drive A

Beispielsequenz:

PC → ELO	**ELO → PC**	**Beschreibung**
!GX1↵	#134↵	Der zurückgelesene Wert zeigt, dass sowohl das **RUN**-Flag, wie auch die **FREEZE**-Flags für die Motoren B und C gesetzt sind.



IP

IP	Inkrementiere Sollposition (relative Positionierung) L S
Richtung	**ASCII-Sequenz**
PC → ELO	‘!IP’ + *posinc*[*sint16*]+ *drive*[*drivecode*] + 0x0D
ELO → PC	‘#’ + *ack*[*ackcode*] + 0x0D

Dieser Befehl inkrementiert die Sollposition des Motors *drive* um den Betrag *posinc*. Diser Befehl darf nur im Zustand **RUN** aufgerufen werden. Die Auflösung der Position kann mit dem Befehl **PI** erfragt werden. Die inkrementierte Position darf minimal -32256 und maximal 32256 betragen.

Beispielsequenz:

PC → ELO	ELO → PC	Beschreibung
!IP100A↵	#↵	Inkrementiert die Sollposition um 100 Einheiten.

MH

MH	Verschiebe Referenzposition L S
Richtung	**ASCII-Sequenz**
PC → ELO	‘!MH’ + *posinc*[sint16] + *drive*[*drivecode*] + 0x0D
ELO → PC	‘#’ + *ack*[*ackcode*] + 0x0D

Dieser Befehl verschiebt die Referenzposition des Motors *drive* um den Wert *posinc*. Dieser Befehl macht nur in wenigen Fällen Sinn und sollte mit Vorsicht eingesetzt werden. Dieser Befehl darf nur im Zustand **RUN** ausgeführt werden. Achtung: die Minimal- und Maximalpositionen werden nicht mitverschoben!

PI

PI	Lese Positionsausflösung L S M
Richtung	**ASCII-Sequenz**
PC → ELO	‘!PI’ + *drive*[*drivecode*] + 0x0D
ELO → PC	‘#’ + *posinc*[*uint32*] + 0x0D

Dieser Befehl erfragt die aktuelle Positionsauflösung. Die Werte werden dabei je nach gewähltem Motortyp in verschiedenen Einheiten zurückgegeben:

Motortyp	Einheit
LinMot®	1 pm (10^{-12}m)
Schrittmotor	1/256 Schritt
Magnet	1 µA

Beispielsequenz:

PC → ELO	ELO → PC	Beschreibung
!PIA↵	#19531250 ↵	Der Rückgabewert von ‘19531250’ bedeutet, dass die Positionsauflösung 19.53µm beträgt (wenn als Motortyp beim Motor ‘A’ ‘LinMot’ gewählt ist). Der Befehl ‘!IP1000A’ wird daher den Motor ‘A’ um 1000 * 19.53µm bewegen.

PV

PV Lese Protokollversion	E
Richtung	**ASCII-Sequenz**
PC → ELO	'!PV' + *elo*$_{[elocode]}$ + 0x0D
ELO → PC	'#' + *version*$_{[uint16]}$ + 0x0D

Dieser Befehl erfragt die aktuelle Versionsnummer des Protokolls. Die im Release 1.3 implementierten Befehle entsprechen der Protokollversion 2. Die in der Protokollversion 2 implementierten Befehle sind eine Obermenge von den in der Protokollversion 1 (Release 1.2) implementierten Befehlen.

Beispielsequenz:

PC → ELO	ELO → PC	Beschreibung
!PV1↵	#2↵	Erfragt die aktuelle Protokollversion.

RA

RA Starte Kurve von aktueller Position	L S M
Richtung	**ASCII-Sequenz**
PC → ELO	'!RA' + *curve*$_{[uint16]}$+ *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *ack*$_{[ackcode]}$ + 0x0D

Dieser Befehl startet ein abgelegtes Bewegungsprofil von der aktuellen Sollposition aus. Der Parameter *curve* definiert dabei das Bewegungsprofil, welches gestartet werden soll. Der zulässige Bereich geht von 0 bis 63, wobei 0 für ein leeres Bewegungsprofil steht. Dieser Befehl darf nur im Zustand **RUN** ausgeführt werden.

Wenn der 'RC'-Befehl gesendet wird, während bereits ein anderes Bewegungsprofil abgefahren wird, wird sofort das neue Profil gestartet.

Achtung: Dieser Befehl ändert den Parameter Curve Position Offset.

RC

RC Starte Kurve	L S M
Richtung	**ASCII-Sequenz**
PC → ELO	'!RC' + *curve*$_{[uint16]}$+ *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *ack*$_{[ackcode]}$ + 0x0D

Dieser Befehl startet ein abgelegtes Bewegungsprofil. Der Parameter *curve* definiert dabei das Bewegungsprofil, welches gestartet werden soll. Der zulässige Bereich geht von 0 bis 63, wobei 0 für ein leeres Bewegungsprofil steht. Dieser Befehl darf nur im Zustand **RUN** ausgeführt werden.

Wenn der 'RC'-Befehl gesendet wird, während bereits ein anderes Bewegungsprofil abgefahren wird, wird sofort das neue Profil gestartet.



RD

RD Lese Datenwort	E
Richtung	**ASCII Sequenz**
PC → ELO	'!RD' + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + data$_{[uint16]}$ + 0x0D

Dieser Befehl liest ein Datenwort (16 Bit) von der Adresse, definiert mit den Befehlen AO und AS. Mit drive wird der Controller adressiert.

Achtung: Die Adresse wird bei Verwendung von den Befehlen WS und RE verändert.

Wert	Min	Max
data$_{[uint16]}$	0	65535

RE

RE Lese Datenwort mit Adressinkrementierung	E
Richtung	**ASCII Sequenz**
PC → ELO	'!RE' + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + data$_{[uint16]}$ + 0x0D

Dieser Befehl liest ein Datenwort (16 Bit) von der Adresse, definiert mit den Befehlen AO und AS. Nach dem Lesen wird die Adresse automatisch um 2 inkrementiert. Mit drive wird der Controller adressiert.

Wert	Min	Max
data$_{[uint16]}$	0	65535

Typische Sequenz:

PC → ELO	ELO → PC	Beschreibung
!REA↵	#11225↵	Liest den Wert 11225 von der vorgängig definierten Adresse.

RP

RP Definiere neue Istposition	L S
Richtung	**ASCII-Sequenz**
PC → ELO	'!RP' + *pos*$_{[sint16]}$ + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *ack*$_{[ackcode]}$ + 0x0D

Dieser Befehl setzt die Istposition des Motors *drive* auf den Wert *pos*. Er macht nur in wenigen Fällen Sinn und sollte mit Vorsicht eingesetzt werden. Achtung: die Minimal- und Maximalpositionen werden nicht mitverschoben! Dieser Befehl darf nur im Zustand **RUN** ausgeführt werden.

Wert	Min	Max
pos$_{[sint16]}$	-32256	32256

RQ

RQ Setze lSollposition auf Istposition	L S
Richtung	**ASCII-Sequenz**
PC → ELO	‘!RQ’ + *drive*[*drivecode*] + 0x0D
ELO → PC	‘#’ + *ack*[*ackcode*] + 0x0D

Dieser Befehl setzt die Sollposition auf die Istposition. Dieser Befehl wir z.B. gebraucht, wenn beim löschen des Freeze der Motor die Bewegung nicht mehr zu Ende fahren soll, oder wenn der Motor stromlos war und nur bei Bestromung keine Bewegung machen darf.

SA

SA Setze maximale Beschleunigung	L S
Richtung	**ASCII-Sequenz**
PC → ELO	‘!SA’ + *acc*[*uint16*] + *drive*[*drivecode*] + 0x0D
ELO → PC	‘#’ + *ack*[*ackcode*] + 0x0D

Dieser Befehl setzt die maximale Beschleunigung des Motors *drive* auf den Wert *acc*. Die Auflösung der Beschleunigung kann mit dem Befehl **AI** erfragt werden.

Wert	Min	Max
acc[*uint16*]	1	1536

SC

SC Setze maximalen Strom (Kraft)	L S M
Richtung	**ASCII-Sequenz**
PC → ELO	‘!SC’ + *curr*[*uint16*] + *drive*[*drivecode*] + 0x0D
ELO → PC	‘#’ + *ack*[*ackcode*] + 0x0D

Dieser Befehl setzt den maximalen Strom des Motors *drive* auf den Wert *curr*. Die Auflösung des Stroms kann mit dem ‘CI’-Befehl erfragt werden.

Wert	Min	Max
curr[*uint16*]	0	256



SF

SF Setze FREEZE-Flags [L] [S] [M] [E]

Richtung	ASCII-Sequenz
PC → ELO	'!SF' + ('+'|'-') + ($elo_{[elocode]}$ | $drive_{[drivecode]}$) + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt ('+') oder löscht ('-') die **FREEZE**-Flags. Auf jedem Servo Controller existiert ein globales **FREEZE**-Flag, welches die Bewegung von allen Motoren einfriert, sowie vier motorspezifische **FREEZE**-Flags, welche die Bewegung von einzelnen Motoren einfrieren. Ein Motor bewegt sich nur, wenn sowohl das globale **FREEZE**-Flag wie auch sein motorspezifisches **FREEZE**-Flag gelöscht sind.

Beispielsequenz:

PC → ELO	ELO → PC	Beschreibung
!SF+A↵	#↵	Setzt das **FREEZE**-Flag auf dem Motor A.
!SF+1↵	#↵	Setzt das globale **FREEZE**-Flag des Servo Controllers mit der ID 1

SI

SI Setze INIT-Flag [E]

Richtung	ASCII-Sequenz
PC → ELO	'!SI' + ('+'|'-') + $elo_{[elocode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt ('+') oder löscht ('-') das 'INIT'-Flag. Die Bedeutung des 'INIT'-Flags ist im Kapitel "Betriebszustände" auf Seite 11 beschrieben.

SP

SP Setze Sollposition (absolute Positionierung) [L] [S] [M]

Richtung	ASCII-Sequenz
PC → ELO	'!SP' + $pos_{[sint16]}$+ $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt die Sollposition für den Motor *drive* auf den Wert *pos*. Die Auflösung der Position kann mit dem Befehl **PI** abgefragt werden.

Wert	Min	Max
$pos_{[sint16]}$	-32256	32256

SR

SR Setze RUN-Flag [E]

Richtung	ASCII-Sequenz
PC → ELO	'!SR' + ('+'|'-') + $elo_{[elocode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt ('+') oder löscht ('-') das RUN-Flag. Die Bedeutung des RUN-Flags ist im Bedienerhandbuch im Kapitel "Betriebszustände" auf Seite 11 beschrieben.

SS

SS Setze STOP-Flag	E
Richtung	**ASCII-Sequenz**
PC → ELO	'!SS' + ('+'|'-') + $elo_{[elocode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt ('+') oder löscht ('-') das STOP-Flag. Die Bedeutung des STOP-Flags ist im Bedienerhandbuch im Kapitel "Betriebszustände" auf Seite 11 beschrieben.

SV

SV Setze maximale Geschwindigkeit	L S
Richtung	**ASCII-Sequenz**
PC → ELO	'!SV' + $vel_{[uint16]}$ + $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt die maximale Geschwindigkeit des Motors *drive* auf den Wert *vel*. Die Auflösung der Geschwindigkeit kann mit dem Befehl **VI** erfragt werden.

Wert	Min	Max
$vel_{[uint16]}$	6	24576

SW

SW Gehe zu Sollposition von aktueller Position (absolute Positionierung)	L S M
Richtung	**ASCII-Sequenz**
PC → ELO	'!SW' + $pos_{[sint16]}$+ $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt die Sollposition für den Motor *drive* auf den Wert *pos*. Die Auflösung der Position kann mit dem Befehl **PI** abgefragt werden. Im Unterschied zum !SP-Befehl startet der V/A-Limiter bei der aktuellen Position. Dieser Befehl kann z.B. beim Wegfahren nach einem Pressen verwendet werden.

Wert	Min	Max
$pos_{[sint16]}$	-32256	32256



TA

TA	Starte Kurve auf Trigger von aktueller Position L S M
Richtung	**ASCII-Sequenz**
PC → ELO	'!TA' + *curve*[u*int16*]+ *drive*[*drivecode*] + 0x0D
ELO → PC	'#' + *ack*[*ackcode*] + 0x0D

Dieser Befehl startet ein abgelegtes Bewegungsprofil beim nächsten Trigger-Puls von der aktuellen Sollposition. Da der Servo Controller über sehr schnelle Triggereingänge verfügt, kann eine Kurve sehr präzise gestartet werden. Die Kurven werden immer auf die positive Flanke gestartet. Der Parameter *curve* definiert das Bewegungsprofil, welches gestartet werden soll. Der zulässige Bereich geht von 0 bis 63, wobei 0 immer für ein leeres Bewegungsprofil steht. Dieser Befehl darf nur im Zustand **RUN** ausgeführt werden.

Achtung: Dieser Befehl ändert den Parameter Curve Position Offset.

TC

TC	Starte Kurve beim nächsten Triggerpuls L S M
Richtung	**ASCII-Sequenz**
PC → ELO	'!TC' + *curve*[u*int16*]+ *drive*[*drivecode*] + 0x0D
ELO → PC	'#' + *ack*[*ackcode*] + 0x0D

Dieser Befehl startet ein abgelegtes Bewegungsprofil beim nächsten Trigger-Puls. Da der Servo Controller über sehr schnelle Triggereingänge verfügt, kann eine Kurve sehr präzise gestartet werden. Die Kurven werden immer auf die positive Flanke gestartet. Der Parameter *curve* definiert das Bewegungsprofil, welches gestartet werden soll. Der zulässige Bereich geht von 0 bis 63, wobei 0 immer für ein leeres Bewegungsprofil steht. Dieser Befehl darf nur im Zustand **RUN** ausgeführt werden.

TI

TI	Inkrementiere Sollposition beim nächsten Trigger L S
Richtung	**ASCII-Sequenz**
PC → ELO	'!TI' + *posinc*[s*int16*]+ *drive*[*drivecode*] + 0x0D
ELO → PC	'#' + *ack*[*ackcode*] + 0x0D

Dieser Befehl inkrementiert die Sollposition des Motors *drive* um den Betrag *posinc* beim nächsten Triggerpuls. Die Bewegung wird immer auf die positive Flanke gestartet. Die inkrementierte Position darf minimal -32256 und maximal 32256 betragen. Dieser Befehl darf nur im Zustand **RUN** ausgeführt werden.

TP

TP Setze Sollposition beim nächsten Trigger	L S M
Richtung	**ASCII-Sequenz**
PC → ELO	'!TP' + $pos_{[sint16]}$+ $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt die Sollposition des Motors *drive* auf den Wert *pos* beim nächsten Triggerpuls. Da der Servo Controller über sehr schnelle Triggereingänge verfügt, kann eine Bewegung sehr präzise gestartet werden. Die Bewegung wird immer auf die positive Flanke gestartet. Dieser Befehl darf nur im Zustand **RUN** ausgeführt werden.

Wert	Min	Max
$pos_{[sint16]}$	-32256	32256

TW

TW Gehe zu Sollposition von aktueller Position beim nächsten Trigger	L S M
Richtung	**ASCII-Sequenz**
PC → ELO	'!TW' + $pos_{[sint16]}$+ $drive_{[drivecode]}$ + 0x0D
ELO → PC	'#' + $ack_{[ackcode]}$ + 0x0D

Dieser Befehl setzt die Sollposition des Motors *drive* auf den Wert *pos* beim nächsten Triggerpuls. Da der Servo Controller über sehr schnelle Triggereingänge verfügt, kann eine Bewegung sehr präzise gestartet werden. Die Bewegung wird immer auf die positive Flanke gestartet. Dieser Befehl darf nur im Zustand **RUN** ausgeführt werden. Im Unterschied zum !TP-Befehl startet der V/A-Limiter bei der aktuellen Position. Dieser Befehl kann z.B. beim Wegfahren nach einem Pressen verwendet werden.

Wert	Min	Max
$pos_{[sint16]}$	-32256	32256



VI

VI	Lese Geschwindigkeitsinkrement L S
Richtung	**ASCII-Sequenz**
PC → ELO	'!VI' + *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *velres*$_{[uint32]}$ + 0x0D

Dieser Befehl erfragt die Geschwindigkeitsauflösung. Die Werte werden dabei, je nach gewähltem Motortyp, in verschiedenen Einheiten zurückgegeben:

Motortyp	Einheit
LinMot®	1nm/s
Schrittmotor	2^{-16} Schritt/s
Magnet	0

Beispielsequenz:

PC → ELO	ELO → PC	Beschreibung
!VIA↵	#190735↵	Erfragt aktuelle Geschwindigkeitsauflösung von Motor A. Der zurückgegebene Wert entspricht 0.190735mm/s, da der Motor A in diesem Beispiel als *LinMot®* konfiguriert war.

WR

WR	Schreibe Datenwort E
Richtung	**ASCII Sequenz**
PC → ELO	'!WR' + data$_{[uint16]}$+ *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *ack*$_{[ackcode]}$ + 0x0D

Dieser Befehl schreibt ein Datenwort (16 Bit) an die Adresse, welche vorgängig mit AO oder AS definiert wurde. Mit drive wird der Controller adressiert.

Achtung: Die Adresse wird bei Verwendung von den Befehlen WS und RE verändert

Wert	Min	Max
data$_{[uint16]}$	0	65535

WS

WS	Schreibe Datenwort mit Adressinkrementierung E
Richtung	**ASCII Sequenz**
PC → ELO	'!WS' + data$_{[uint16]}$+ *drive*$_{[drivecode]}$ + 0x0D
ELO → PC	'#' + *ack*$_{[ackcode]}$ + 0x0D

Dieser Befehl schreibt ein Datenwort (16 Bit) an die Adresse, welche vorgängig mit AO oder AS definiert wurde. Nach dem Zugriff wird die Adresse automatisch um 2 inkrementiert. Mit drive wird der Controller adressiert.

Wert	Min	Max
data$_{[uint16]}$	0	65535

ZD

ZD Setze internen Positionszähler auf den Wert 0 L S	
Richtung	**ASCII-Sequenz**
PC → ELO	'!ZD' + *drive*[*drivecode*] + 0x0D
ELO → PC	'#' + ack[*ackcode*] + 0x0D

Dieser Befehl setzt den internen Positionszähler auf den Wert 0 und verschiebt somit auch die 'Home'-Position. Dieser Befehl macht nur in einigen Fällen Sinn und sollte mit Vorsicht eingesetzt werden. Achtung: die Minimal- und Maximalpositionen werden nicht mitverschoben! Dieser Befehl darf nur im Zustand **RUN** ausgeführt werden.



8.6 Beispielsequenz

Die folgende Beispielsequenz gibt einen Überblick über die Möglichkeiten des ASCII-Protokolles. Das '↵'-Zeichen steht dabei für den hexadezimalen Wert 0x0D. Damit diese Sequenz nachgespielt werden kann, müssen an den Motorensteckern 'Mot A' und 'Mot B' je ein *LinMot®* angeschlossen und konfiguriert sein. Zudem muss mindestens ein Bewegungsprofil auf dem Servo Controller abgelegt sein.

Beispielsequenz

PC → ELO	ELO → PC	Beschreibung
`!PV1↵`	`#2↵`	Erfragen der Protokollversion des Servokontrollers.
`!GS1↵`	`#W↵`	Erfragen des Betriebszustandes von Kontroller '1'.'W' bedeutet "Wait for Disable".
`!SR-1↵`	`#↵`	Löschen des RUN Request Bits.
`!SI-1↵`	`#↵`	Löschen des INIT Request Bits.
`!GS1↵`	`#D↵`	Erfragen des Betriebszustandes von Kontroller '1'. 'D' bedeutet "Disable".
`!SI+1↵`	`#↵`	Setzen des INIT request Bits. Die Initialisierung startet.
`!EX1↵`	`#1024↵`	Warten bis "Init Not Done flag" gelöscht ist. Der zurückgegebene Wert definiert den Zustand "INIT". Dieser Befehl kann soll wiederholt werden, bis die Initialisierung abgeschlossen ist.
`!SR+1↵`	`#W↵`	Setze RUN Request Bit.
`!SI-1↵`	`#↵`	Löschen des INIT Request Bits.
`!GS1↵`	`#R↵`	Erfragen des Zustandes. 'R' bedeutet Zustand RUN.
`!CI1↵`	`#23438↵`	Erfragen des Strominkrementes von Servokontroller 1. Der zurückgegebene Wert bedeutet 23.439 mA.
`!SC43B↵`	`#↵`	Setzen des Maximalstromes für Motor B auf 1 A.
`!SC64A↵`	`#↵`	Setzen des Maximalstromes für Motor A auf 1.5 A.
`!ZDB↵`	`#↵`	Die Position von Motor B auf 0 setzen.
`!PIA↵`	`#19531250↵`	Erfragen des Positionsinkrementes von Motor A. Der zurückgegebene Wert bedeutet 19.53125 μm.
`!SP2560B↵`	`#↵`	Setzen der Sollposition von Motor B auf 50 mm.
`!RC1A↵`	`#↵`	Starte Kurvennummer 1 von Motor A.
`!GPB↵`	`#2560↵`	Erfragen der aktuellen Position von Motor B.

8.7 Referenztabelle: Status- und Fehlermeldungen

Quittierungs-Codes (ackcode)	
Code	**Beschreibung**
‘’	Okay, kein Fehler.
‘E50’	Gesendetes Kommando kann nicht im Zustand **WAIT FOR DISABLE** ausgeführt werden.
‘E51’	Gesendetes Kommando kann nicht im Zustand **DISABLE** ausgeführt werden.
‘E52’	Gesendetes Kommando kann nicht im Zustand **INIT** ausgeführt werden.
‘E53’	Gesendetes Kommando kann nicht im Zustand **ERROR** ausgeführt werden.
‘E54’	Gesendetes Kommando kann nicht im Zustand **RUN** ausgeführt werden.
‘E55’	Interner Fehler.
‘E56’	Falscher Controller- oder Motor gewählt. Diese Fehlermeldung wird gesendet, falls sich der Servo Controller im RS232-Modus befindet und ein Befehl einen Controller oder einen Motor adressiert, welche in diesem Modus nicht erlaubt ist. Im RS232-Modus dürfen nur der Controller ‘1’ und die Motoren ‘A’ … ‘D’ verwendet werden!
‘E57’	Der selektierte Motor ist nicht vorhanden oder es ist ‘No drive’ als Motortyp gewählt.
‘E58’	Unbekannter Befehl.
‘E59’	Falsches Befehlsformat.
‘E60’	Vorzeichenfehler.
‘E61’	Es wurde versucht das **RUN**, **INIT**, **FREEZE** oder **STOP** Flag zu setzen, obwohl im Parameter **IO Configuration** eines dieser Flags gesetzt wurde. Wenn eines der Flags gesetzt ist, wird der Zustand des Flags alleine durch den digitalen Eingang auf dem Servo Controller bestimmt.
‘E62’	Das gewählte Bewegungsprofil ist auf dem Servo Controller nicht vorhanden.
‘E63’	Es wurde versucht ein Bewegungsprofil zu starten, welches nicht für den aktuellen Motorentyp erstellt wurde.
‘E64’	Wertebereich überschritten.
‘E65’	Befehl ist zu lange.
‘E66’	Der Motor ist nicht im Betriebsmodus ‚Serial‘. Im Parameter-Inspector unter **\Drives\Drive X\Set Value Generation\Run Mode** muss **Serial** gewählt sein.
‘E67’	Der Befehl kann nicht auf Slave-Motoren angewendet werden.
‘E68’	Der Befehl kann nicht mit dem gewählten Motorentyp ausgeführt werden.

Status-Codes (statecode)	
Code	**Beschreibung**
‘W’	Servo Controller ist im Zustand **WAIT FOR DISABLE**
‘R’	Servo Controller ist im Zustand **RUN**
‘I’	Servo Controller ist im Zustand **INIT**
‘D’	Servo Controller ist im Zustand **DISABLE**
‘E’ + *syserr*	Servo Controller ist im Zustand **ERROR**. Es ist ein Systemfehler aufgetreten. Die Kodierung von *syserr* ist in der Systemfehlertabelle weiter unten abgelegt.
‘E’ + *driveerr* + ‘A’	Servo Controller ist im Zustand **ERROR**. Es ist ‘Motor A’-Fehler aufgetreten. Die Kodierung von *driveerr* ist in der Motorfehlertabelle weiter unten abgelegt.
‘E’ + *driveerr* + ‘B’	Servo Controller ist im Zustand **ERROR**. Es ist ‘Motor B’-Fehler aufgetreten.
‘E’ + *driveerr* + ‘C’	Servo Controller ist im Zustand **ERROR**. Es ist ‘Motor C’-Fehler aufgetreten.
‘E’ + *driveerr* + ‘D’	Servo Controller ist im Zustand **ERROR**. Es ist ‘Motor D’-Fehler aufgetreten.



Systemfehler-Codes (syserr)	
Code	**Beschreibung**
‘2’	Versorgungsspannung für den Leistungsteil ist zu tief.
‘3’	Versorgungsspannung für den Leistungsteil ist zu hoch.
‘4’	Versorgungsspannung für den Signalteil ist zu tief.
‘5’	Versorgungsspannung für den Signalteil ist zu hoch.
‘6’	Servo Controller ist zu heiss.
‘7’	Interne 12V im Servo Controller liegen nicht an.

Motorfehler-Codes (driveerr)	
Code	**Beschreibung**
‘1’	Motor ist überlastet.
‘2’	Motor ist zu heiss.
‘3’	Schleppfehler ist aufgetreten.
‘4’	Läufer des Motors fehlt.
‘6’	Während der Initialisierung ist ein Fehler aufgetreten.
‘7’	Falscher Motortyp konfiguriert oder Motor defekt.
‘8’	Bewegungsprofil für den Motor fehlt oder gewähltes Bewegungsprofil ist mit dem Motortyp nicht kompatibel.
‘9’	Überstom im Motorkanal detektiert. Mögliche Ursachen: Verkabelungsfehler, Kurzschluss auf den Phasen...
‘10’	Übertemperatur im Servo Controller detektiert. Mögliche Ursachen: Motor zieht zu viel Strom, unzureichende Controllerkühlung, ...
‘11’	Sicherung auf AGND oder 5VDC des Motorkanals defekt. Mögliche Ursachen: Kurzschluss zwischen Motorphasen und Signalleitungen.

8.8 Referenztabelle: Positionsinkrement

Die folgende Tabelle listet die bei der Benutzung des **PI**-Befehls zurückgesandten Werte auf. Diese Werte können sich jedoch bei einer zukünftigen ASCII-Protokollversion ändern. Wenn immer möglich sollte daher das Positionsinkrement mit dem **PI**-Befehl abgefragt werden.

Motortyp	Inkrement
LinMot®	19.53125µm
Stepper	1/8 Step
Magnet	23.438mA

8.9 Referenztabelle: Geschwindigkeitsinkrement

Die folgende Tabelle listet die bei der Benutzung des **VI**-Befehls zurückgesandten Werte auf. Diese Werte können sich jedoch bei einer zukünftigen

ASCII-Protokollversion ändern. Wenn immer möglich sollte daher das Geschwindigkeitsinkrement mit dem **VI**-Befehl abgefragt werden.

Motortyp	Inkrement
LinMot®	190.735µm/s
Stepper	0.081469 Steps/s
Magnet	0

8.10 Referenztabelle: Beschleunigungsinkrement

Die folgende Tabelle listet die bei der Benutzung des ‘AI’-Befehls zurückgesandten Werte auf. Diese Werte können sich jedoch bei einer zukünftigen ASCII-Protokollversion ändern. Wenn immer möglich sollte daher das Beschleunigungsinkrement mit dem ‘AI’-Befehl abgefragt werden.

Motortyp	Inkrement
LinMot®	238.419mm/s^2
Stepper	47.6836 Steps/s^2
Magnet	-

8.11 Referenztabelle: Strominkrement

Die folgende Tabelle listet die bei der Benutzung des ‘CI’-Befehls zurückgesandten Werte auf. Diese Werte können sich jedoch bei einer zukünftigen ASCII-Protokollversion ändern. Wenn immer möglich sollte daher das Strominkrement mit dem ‘CI’-Befehl abgefragt werden.

Servo Controller	Inkrement
Ex00 / Ex000	23.438mA

8.12 Referenztabelle: Motorbezeichner

Die Motorbezeichner identifizieren einen Motor in einer Systemlösung in welcher das ASCII-Protokoll unter RS485 eingesetzt wird.

Controller ID	Motor A	Motor B	Motor C	Motor D
1	A	B	C	D
2	E	F	G	H
3	I	J	K	L
4	M	N	O	P
5	Q	R	S	T
6	U	V	W	X



A. Kompatibilität zu alten Releases

Damit ein problemloser Übergang zu Release 1.3 von *LinMot®* Talk sichergestellt ist, können Konfigurations- und Kurvendateien, die mit alten Versionen erstellt wurden, in Release 1.3 von *LinMot®* Talk importiert werden. Es werden die Dateien von Release 1.0, 1.1 sowie 1.2 unterstützt. Beim Importieren von ‘alten’ Konfigurationen ist eine Konvertierung nötig. Diese Konvertierung wird vom **Parameter Inspector** automatisch vorgenommen. Um Problemen vorzubeugen, sollte die so erstellte neue Konfiguration sorgfältig überprüft werden.

Servo Controller die den SW-Release 1.0 oder 1.1 geladen haben können nicht mit *LinMot®* Talk Software Release 1.3 betrieben werden. Mit Release 1.3 von *LinMot®* Talk können Servo Controller mit dem SW-Release 1.2 und 1.3 betrieben werden. Es ist ohne Probleme möglich, die *LinMot®* Talk Releases 1.0, 1.1, 1.2 und 1.3 gleichzeitig auf einem PC zu installieren. Mit einer solchen Konfiguration können alle Servo Controller bedient werden. Es muss jedoch darauf geachtet werden, dass immer nur eine Version von *LinMot®* Talk gleichzeitig gestartet wird. Alle SW-Releases sind auf der *LinMot®* Homepage unter der WWW-Adresse **http://www.linmot.com** zu finden.

	importiert Konfigurationen von				**konfiguriert Servo Controller mit**			
	R1.0	R1.1	R1.2	R1.3	R1.0	R1.1	R1.2	R1.3
LinMot® Talk R1.0	✘				✘			
LinMot® Talk R1.1	✘	✘				✘		
LinMot® Talk R1.2	✘	✘	✘				✘	
LinMot® Talk R1.3	✘	✘	✘	✘			✘	✘

Tabelle 1-1: Kombatibilität von *LinMot®* Talk Software

Beachte: Der sicherste Weg um eine Konfiguration und Kurven von einem SW-Release zum anderen zu übertragen ist indem die Konfiguration in der gleichen *LinMot®* Talk gespeichert wird, wie der Quellkontroller Firmware geladen hat und beim Importieren das *LinMot®* Talk ebenfalls der Version der Zielkontrollerfirmware entspricht.

Beachte: Eine neuere Hardwareversion oder -Revision bedingt prinzipiell auch eine neuere Firmwareversion. Die neueste Hardware wird immer von den aktuellsten Firmware-Versionen 1.1, 1.2 und 1.3 unterstützt.

B. Service / Fehleranzeige

Fehleranzeige Version 2 Kontroller

Beim Auftreten eines Fehlers wird dieser durch Blinken von mindestens einer der vier Leuchtdioden (LED) auf der Frontseite des Servo Controller angezeigt.

Ready LED

Die Ready LED leuchtet, sobald die Spannungsversorgung für den Prozessor gewährleistet ist und das System korrekt aufgestartet hat. Eine Kommunikation mit dem PC über die serielle Schnittstelle ist dann möglich.

In der folgenden Tabelle sind die Blinkkodes und die dazugehörenden Fehlermeldungen aufgeführt:

Fault LED	Stat A	Stat B	Beschreibung
● ~3Hz	aus	aus	Systemfehler HW: Hardwarefehler im Servo Controller.
2x● 1Hz			Systemfehler SW1: Softwarefehler im Servo Controller. Die Systemsoftware wurde nicht fehlerfrei geladen.
● ~1Hz			Systemfehler SW2: Softwarefehler im Servo Controller. Die Systemsoftware wurde nicht fehlerfrei geladen.
ein	● ~2Hz	● ~2Hz	Generischer Fehler: Die genaue Fehlermeldung kann mit dem Error-Inspector angezeigt werden.
ein	▬	2x●	Die Versorgungsspannung für den Leistungsteil ist zu tief.
		3x●	Die Versorgungsspannung für den Leistungsteil ist zu hoch.
		4x●	Die Versorgungsspannung für den Signalteil ist zu tief.
		5x●	Die Versorgungsspannung für den Signalteil ist zu hoch.
		6x●	Der Servo Controller ist zu heiss.
	1x● : Mot A 2x● : Mot B 3x● : Mot B 4x● : Mot D	1x●	Der Motor ist überlastet.
		2x●	Der Motor ist zu heiss.
		3x●	Schleppfehler
		4x●	Der Läufer des Motors fehlt.
		6x●	Die Initialisierung konnte nicht erfolgreich abgeschlossen werden.
		7x●	Falscher Motortyp konfiguriert oder Motor defekt.
		8x●	Eine Sollwertkurve für den Motor fehlt / Kurve ist mit dem Aktor-Typ nicht kompatibel

Tabelle 2-1: Fehleranzeige LEDs bei Version 2 Kontrollern

Legende			
on/ein	LED leuchtet dauernd	●	LED leuchtet kurz (ca. ¼ s)
off/aus	LED ist dunkel	▬	LED leuchtet lange (ca. 1½ s)



Fehleranzeige Version 3 Kontroller

Beim Auftreten eines Fehlers bei einem Version 3 Kontroller wird die Fehlernummer auf der 7-Segment Anzeige dargestellt. Die folgende Tabelle zeigt die möglichen Fehlercodes und deren Bedeutung auf:

Fehlercode	Beschreibung
E0001	Fehlender oder ungültiger Parameterbaum
E0002	Fehlende oder ungültige Applikation
E0003	Kontrollertyp nicht unterstützt
E0004	MT Befehlsinterface nicht verfügbar
E0005	Timer Watch Dog Fehler
E0006	Trap Klasse A Fehler
E0007	Trap Klasse B Fehler
E0008	Kein Master für einen Slave Motor definiert
E0009	Kein externer Sensor definiert
E000A	Externer Sensor nicht erlaubt auf Kanal D.
E000B	Die Applikation benötigt einen MT-Kontroller
E000C	Das Noise Dead Band ist bei dieser Geräteversion nicht unterstützt (muss auf 0mm gesetzt werden).
E0010	Zwischenkreisspannung für Leistung zu tief
E0011	Zwischenkreisspannung für Leistung zu hoch
E0012	Zwischenkreisspannung Signalteil zu tief
E0013	Zwischenkreisspannung Singalteil zu hoch
E0014	Elektronikfehler
E0015	Interne 12V-Spannung fehlt.
E0101	Motor A: Berechnete Temperatur zu hoch.
E0102	Motor A: Übertemperaturfehler
E0103	Motor A: Schleppfehler
E0104	Motor A: Läufer fehlt
E0106	Motor A: Fehler beim Initialisieren
E0107	Motor A: Falscher Motortyp
E0108	Motor A: Kurvenfehler (Typ oder Nummer falsch)
E0109	Motor A: Überstrom
E010A	Motor A: Übertemperatur auf Motorkanal im Kontroller
E010B	Motor A: AGND oder 5VDC Sicherung defekt
E0201	Motor B: Berechnete Temperatur zu hoch.
E0202	Motor B: Übertemperaturfehler
E0203	Motor B: Schleppfehler
E0204	Motor B: Läufer fehlt
E0206	Motor B: Fehler beim Initialisieren
E0207	Motor B: Falscher Motortyp
E0208	Motor B: Kurvenfehler (Typ oder Nummer falsch)
E0209	Motor B: Überstrom
E020A	Motor B: Übertemperatur auf Motorkanal im Kontroller
E020B	Motor B: AGND oder 5VDC Sicherung defekt

Fehlercode	Beschreibung
E0301	Motor C: Berechnete Temperatur zu hoch.
E0302	Motor C: Übertemperaturfehler
E0303	Motor C: Schleppfehler
E0304	Motor C: Läufer fehlt
E0306	Motor C: Fehler beim Initialisieren
E0307	Motor C: Falscher Motortyp
E0308	Motor C: Kurvenfehler (Typ oder Nummer falsch)
E0309	Motor C: Überstrom
E030A	Motor C: Übertemperatur auf Motorkanal im Kontroller
E030B	Motor C: AGND oder 5VDC Sicherung defekt
E0401	Motor D: Berechnete Temperatur zu hoch.
E0402	Motor D: Übertemperaturfehler
E0403	Motor D: Schleppfehler
E0404	Motor D: Läufer fehlt
E0406	Motor D: Fehler beim Initialisieren
E0407	Motor D: Falscher Motortyp
E0408	Motor D: Kurvenfehler (Typ oder Nummer falsch)
E0409	Motor D: Überstrom
E040A	Motor D: Übertemperatur auf Motorkanal im Kontroller
E040B	Motor D: AGND oder 5VDC Sicherung defekt
$FFEC	RAM Fehler
E1001	RTS: State ist zu lang
E1002	RTS: Kontrollerversion nicht unterstützt
E1003	RTS: Falsche Firmware
E1004	RTS: Kein Script gefunden
E1005	RTS: Ungültiger Befehl
E8000	Profibus DP: Verbindung zum Master verloren
E8001	Profibus DP: Ungültige Adresse
E8002	Profibus DP: Datenbereichsfehler
E8003	Profibus DP: Ungültige Konfiguration von der SPS
E8100	DN: Applikation benötigt DeviceNet Kontroller
E8101	DN: DeviceNet MACID bereits benutzt
E8104	DN: Prozessorfrequenz nicht unterstützt
E8108	DN: Unbekannter Befehl
E8110	DN: Befehl hat keine Drive-Spezifikation
E8111	DN: Drive ist nicht Master
E8112	DN: Drive ist nicht im seriellen Modus
E8118	DN: Bereichsfehler



Fehlercode	Beschreibung
E8120	DN: Encoder existiert nicht
E8121	DN: Encoder ist in SSI-Modus
E8130	DN: Kurve existiert nicht
E8131	DN: Falscher Kurventyp
E8132	DN: Eine Kurve ist noch am Laufen
E8140	DN: Undefinierter CAN Fehler
E8141	DN: CAN Stuff Fehler
E8142	DN: CAN Form Fehler
E8143	DN: CAN Acknowledge Fehler
E8144	DN: CAN Bit1 Fehler
E8145	DN: CAN Bit0 Fehler
E8146	DN: CAN CRC Fehler
E8147	DN: CAN Message verloren
E8148	DN: CAN BOFF Fehler
E8200	CO: Applikation benötigt CANopen Kontroller
E8201	CO: Ungültige Adresse
E8202	CO: Dantenbereichsfehler
E8203	CO: Drive ist nicht im seriellen Modus
E8210	CO: Bus Fehler
E8218	CO: Undefinierter CAN Fehler
E8219	CO: CAN Stuff Fehler
E821A	CO: CAN Form Fehler
E821B	CO: CAN Acknowledge Fehler
E821C	CO: CAN Bit1 Fehler
E821D	CO: CAN Bit0 Fehler
E821E	CO: CAN CRC Fehler

Tabelle 2-2: Fehleranzeige bei Version 3 Kontrollern



C. Wartung der Servo Controller

Die Servo Controller besitzen keine vom Anwender wartbare Teile. Im Normalbetrieb ist ein Öffnen der Servo Controller nicht erforderlich. Damit die von den Servo Controller produzierte Wärme problemlos abgeführt werden kann, sollte der Kühlkörper regelmässig von Staub und anderen Ablagerungen befreit werden.

Sicherungen

Die Versorgungseingänge der Servo Controller sind mit Sicherungen gegen Überströme geschützt. Auf dem Leistungsprint befinden sich zwei Feinsicherungen, je eine für die Signalspeisung und eine für die Leistungsspeisung. Die folgenden Abbildungen zeigen die Lage der Sicherungen:

E100-AT/MT/DP

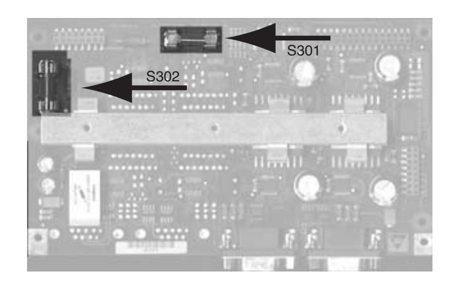


Abbildung C-1: Sicherungen *LinMot® Ex00-AT/MT/DP* Leistungsprint

Sicherung	Typ
S301, Leistungsversorgung	10A träge, ∅ 5mm x 20mm
S302, Signalversorgung	0.5A träge, ∅ 5mm x 20mm

Ex000-AT/MT/DP

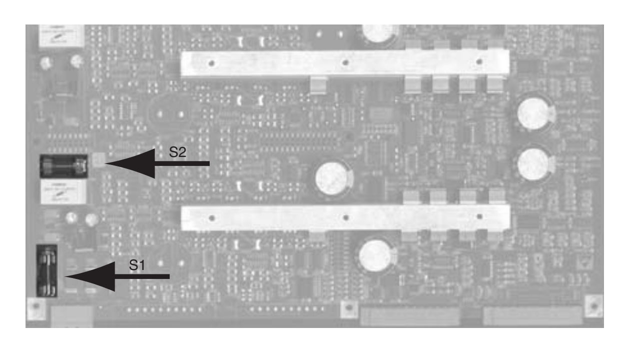


Abbildung C-2: Sicherungen *LinMot® Ex000-AT/MT/DP* Leistungsprint

Sicherung	Typ
S1, Leistungsversorgung	10A träge, ∅ 5mm x 20mm
S2, Signalversorgung	0.5A träge, ∅ 5mm x 20mm



D. Wartung der Motoren

Der nachfolgende Wartungsplan geht von einer 5 Tage-Woche mit einer täglichen Betriebszeit von 8 Stunden aus. Es werden mitteleuropäische Industriebedingungen vorausgesetzt. Bei abweichenden Anforderungen wie etwa starke und permanente Verschmutzung, direkte Sonneneinstrahlung, Betrieb im Freien etc. muss der Wartungsplan verkürzt werden, bis applikationsspezifische Erfahrungswerte vorliegen.

Wartungsplan Standardanwendung

Folgender Wartungsplan sollte bei normalen Bedingungen befolgt werden

Zeitraum	unter 120 Hübe/min	120 bis 360 Hübe/min	über 360 Hübe/min
Inbetriebnahme	Inspektion Schmierung	Inspektion Schmierung	Inspektion Schmierung
alle 3 Monate	- -	Inspektion	Inspektion Schmierung
alle 6 Monate	Inspektion	Inspektion Schmierung	Inspektion Schmierung

Wartungsplan Erstanwendung / erschwerte Bedingungen

Bei Erstanwendungen oder unter erschwerten Bedingungen sollte dieser Wartungsplan befolgt werden:

Zeitraum	unter 120 Hübe/min	120 bis 360 Hübe/min	über 360 Hübe/min
Inbetriebnahme	Inspektion Schmierung	Inspektion Schmierung	Inspektion Schmierung
nach den ersten 8 Std.	Inspektion	Inspektion	Inspektion
nach der ersten Betriebswoche	Inspektion	Inspektion	Inspektion
alle 3 Monate	Inspektion	Inspektion	Inspektion Schmierung
alle 6 Monate	Inspektion Schmierung	Inspektion Schmierung	Inspektion Schmierung

Inspektion

Bei der Inspektion der Antriebe sind folgende Punkte zu überprüfen:

- Ist der Läufer vollständig mit Schmiermittel versehen?
- Ist das Schmiermittel frei von Verschmutzung?
- Lässt sich der Läufer ohne Kraftaufwand bewegen?
- Befindet sich das Anschlusskabel in einwandfreiem Zustand?

Reinigung

Auf keinen Fall dürfen harte Bürsten oder ähnliche Werkzeuge für die Reinigung der *LinMot®* Linearmotoren verwendet werden. Es dürfen keine Reinigungsflüssigkeiten verwendet werden, die Lösungsmittelzusätzte, Petroleum oder vergleichbare Produkte enthalten. Bei der Reinigung sollte folgendermassen vorgegangen werden.

1 Läufer und Stator mit einem weichen Wegwerfpapier reinigen. Falls nötig können Rückstände mit Brennsprit oder Alkohol entfernt werden.

2 Läufer mit Schmiermittel versehen und vorsichtig einführen.

Schmiervorschrift

Das Schmiermittel verbessert die Gleiteigenschaften zwischen der Chromnikkelstahl-Oberfläche des Läufers und dem Kunststoffgleitlager (Basis POM bzw. Delrin). Zusätzlich verhindert das Schmiermittel Korrosionsbildung (Passrost). Die verwendeten Schmiermittel dürfen das Gleitlagermaterial nicht angreifen und müssen bis 100°C temperaturfest sein. Wichtig ist, dass das Schmiermittel auch bei tiefen Temperaturen eine geringe Viskosität aufweist und bei höheren Temperaturen nicht verdampft.

Die folgende Tabelle gibt eine Übersicht über verschiedene Einsatzbereiche und empfohlene Schmiermittel.

Anwendung	Bezeichnung	Vertreiber	Beschreibung
Gleitlager / Lebensmittelindustrie	*LinMot®* Schmiermittel 0150-1950	*LinMot®*	Gleitlagerschmierpaste für alle *LinMot®* Anwendungen mit Langzeitschmierwirkung. USDA H1 zugelassen. Synthetisches Tieftemperatur- und Hochgeschwindigkeitsfett für Wälz-, Gleitlager Ketten und Dichtungen.
	Klybersynth UH1 14-31	Klüber Lubrication	
Gleitlager	Molykote DX	Molykote	Alternative zu *LinMot®*-Fett
Wälzlager	Microlube GBU Y 131	Klüber Lubrication	Für Wälz- und Gleitlager, vorzugsweise bei hohen spezifischen Belastungen sowie bei Einfluss von Feuchtigkeit und Wasser.
Vakuum	Barrierta L55	Klüber Lubrication	Hochtemperaturfett für Laufrollen und Kugellager in Förderanlagen und Backautomaten sowie Wälzlager und Gleitlager in Elektromotoren.

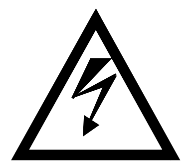
Tabelle D-1: Schmiermittel



E. Mechanische Installation des Controllers

Die *LinMot®* Servo Controller können mit je zwei M5 Schrauben an der Rückwand befestigt werden. Die Befestigungslaschen sind so gestaltet, dass der Servo Controller einfach ein- und ausgebaut werden kann.

Der Servo Controller sollte möglichst in vertikaler Lage montiert werden. Daraus resultiert eine bessere Kühlung. Die *LinMot®* Servo Controller verfügen über eine Temperaturüberwachung die im Falle eines überhitzten Kühlkörpers die Leistungselektronik abschaltet.



ACHTUNG: Bei der Montage der Servo Controller ist zu beachten, dass die Gehäusetemperatur bis zu 60°C betragen kann, am Kühlkörper bis zu 70°. Es muss für eine genügende Wärmeabfuhr am Einbauort gesorgt werden.

Wie bereits im Kapitel über die Stromversorgung erwähnt, muss der Servo Controller zuverlässig geerdet werden.

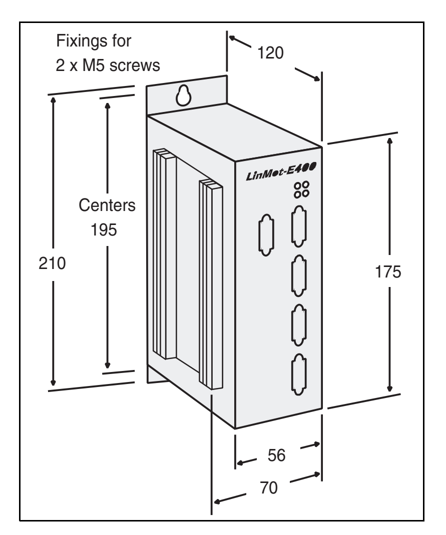


Abbildung E-1: Abmessungen des *LinMot® Serie E100* Servo Controller

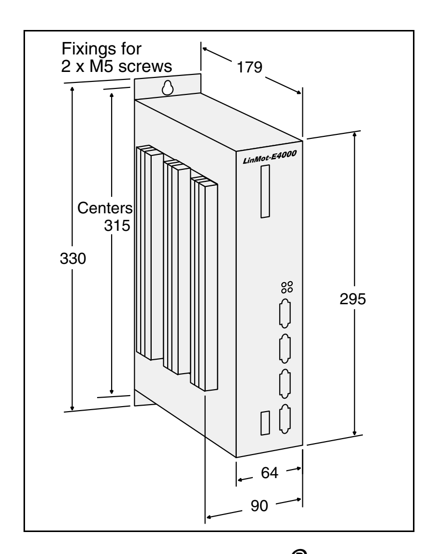


Abbildung E-2: Abmessungen des *LinMot®* Serie *E1000* Servo Controller

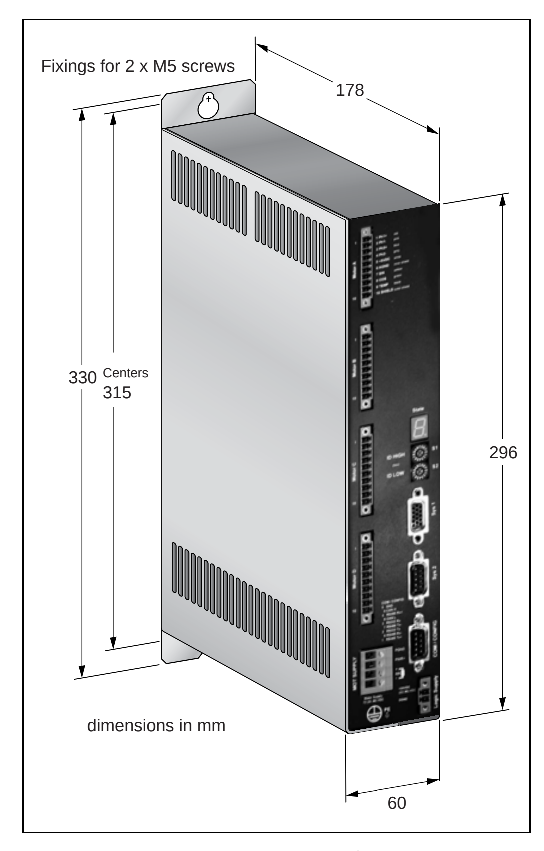


Abbildung E-3: Abmessungen des *LinMot®* Serie *E1001* Servo Controller

.

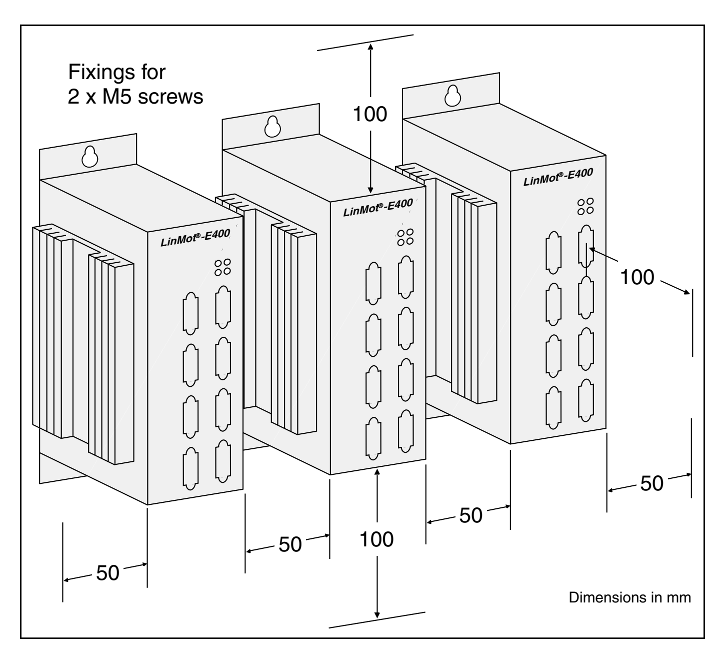


Abbildung E-4: Montageabstände der *LinMot®* *E100* Servo Controller



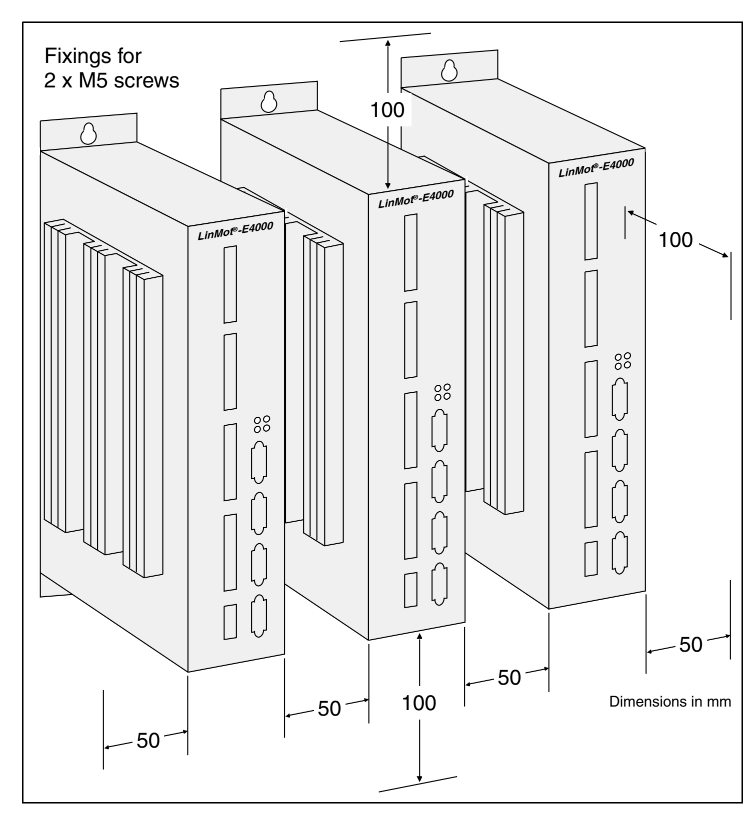


Abbildung E-5: Montageabstände der *LinMot® E1000* Servo Controller

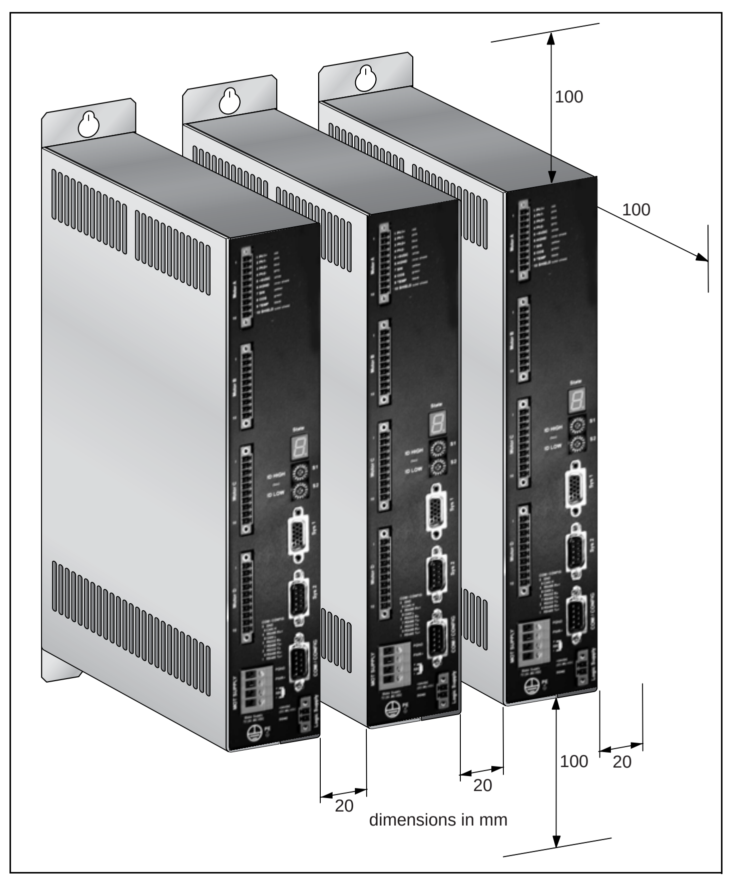


Abbildung E-6: Montageabstände der *LinMot® E1001* Servo Controller



F. Einbau der *LinMot®P* Linearmotoren

Die Linearmotoren der *LinMot® P* Familie besitzen zwischen dem Läufer und dem Stator ein Gleitlager. Die Anforderungen an diese Gleitlagerkonstruktion sind wegen der hohen Dynamik und Geschwindigkeit dieser Antriebe ausserordentlich hoch. Aus diesem Grund müssen die folgenden Punkte unbedingt beachtet werden.

Querbelastung

Eine Querbelastung wirkt sich über den Faktor “Flächenpressung” unmittelbar auf die Belastung des Gleitlagers und somit auf die Lebensdauer des Antriebs aus. Bei der Konstruktion muss darauf geachtet werden, dass der Motor so kleinen Querbelastungen wie möglich ausgesetzt ist. Um eventuell auftretende Querkräfte aufzunehmen, müssen externe Führungen eingesetzt werden.

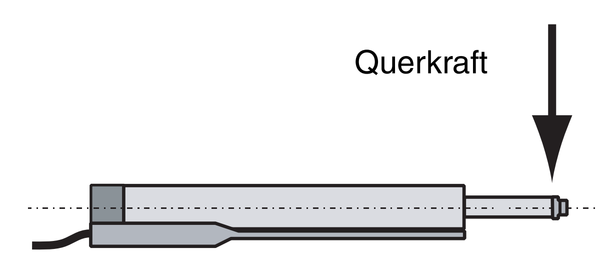


Abbildung F-1: Querkraft zur Bewegungsrichtung

Die Linearmotoren der *LinMot® P* Familie sind ausschliesslich als Antriebselemente konzipiert und dürfen nicht als Lagerelemente eingesetzt werden.

Parallelitätsfehler

Wird ein separat geführtes Maschinenteil mit einem *LinMot® P* Antrieb direkt bewegt, entsteht eine überbestimmte Lageranordnung.

Um die zwangsläufig auftretenden Versatz- und Parallelitätsfehler auszugleichen, muss eine ausgleichsfähige Verbindung, in Abbildung F-2 als gebogene Linie dargestellt, zwischen dem Läufer und dem bewegten Maschinenteil eingebaut werden.

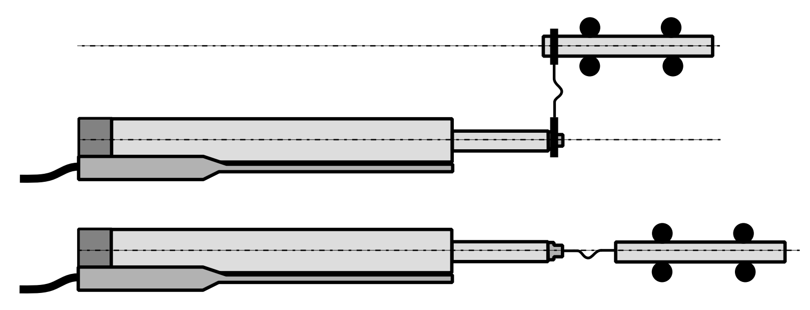


Abbildung F-2: Ausgleich von Parallelfehlern

Einbaulage

Falls ein *LinMot® P* Linearmotor in horizontaler Lage eingebaut wird, sollte er so montiert werden, dass sich die Nut des Stators unten befindet, so wie in obigen Skizzen. Durch diese Einbaulage wird ein Teil des Läufergewichts durch die magnetischen Zugkräfte kompensiert.

Montage des Stators

Der Stator eines *LinMot® P* Linearmotors wird mit einer Klemmverbindung befestigt. Es wird empfohlen die *LinMot®* Flansche der Serie *PF01* oder eine funktionell identische Vorrichtung mit breiter Auflagefläche zu verwenden. Die

Klemmvorrichtung muss einen stabilen Halt der Statoren ermöglichen, darf aber auf keinen Fall den Stator zusammenpressen und damit die Bewegung des Läufers beeinträchtigen.

Bei Verwendung der *LinMot® PF01* Flansche sollte ein Drehmomentschlüssel verwendet und die Angaben im Datenbuch beachtet werden (max. 200 Ncm).

Verbindung mit der Lastmasse

Die Verbindung mit der Lastmasse erfolgt über das Gewinde im Endteil des Läufers. Während der Montage sollte das der Lastmasse zugewandte Endstück mit einem passenden Gabelstück festgehalten werden. Auf keinen Fall darf das empfindliche Läuferrohr oder das gegenüberliegende Läuferende festgeklemmt werden.

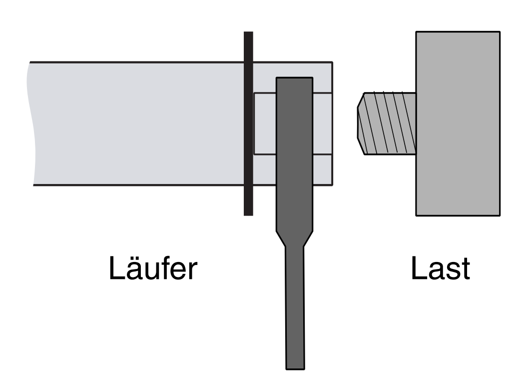


Abbildung F-3: Verbindung mit der Lastmasse

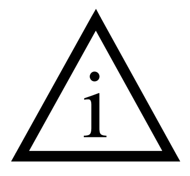
Umgang mit dem Läufer

Die Läufer der *LinMot® P* Motoren müssen ausserordentlich sorgfältig behandelt werden. Bereits kleinste Beschädigungen der Oberfläche bzw. Verkrümmungen des Läufers können zu einer drastischen Reduktion der Lebensdauer führen. Der Läufer ist ein hochgenaues Maschinenelement, welches aus einem dünnen Stahlrohr und Neodym-Magneten aufgebaut ist. Da der Läufer für eine Belastung in Längsrichtung ausgelegt ist, kann bereits das “Anklatschen” an eine Eisenplatte aufgrund der Magnetanziehung und das nachfolgende unsachgemässe Ablösen (radiales Abwuchten!) zu einer bleibenden Beschädigung (Verkrümmung) des Läufers führen. Ein beschädigter Läufer darf keinesfalls weiterverwendet werden, da dies zur Zerstörung des Stators führen kann.

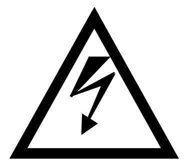
Einführen des Läufers

Bevor der Läufer in den Stator eingeführt wird, ist er mit einem Wegwerfpapier zu reinigen, wobei insbesondere darauf zu achten ist, dass keine Metallspäne an der Läuferoberfläche haften. Danach muss der Läufer gemäss Vorschrift im Kapitel D. eingefettet werden.

Der Läufer muss mit dem Läuferende **ohne Kennrille** voran exakt in Längsrichtung zum Stator eingeführt werden.



Viele nützliche Hinweise zu mechanischen Lösungen finden sich im *LinMot®* Konstruktionshandbuch (Art. Nr. 0150-2215).



ACHTUNG:

- Ein beschädigter Läufer darf auf keinen Fall weiterverwendet werden, da dies zu einer unmittelbaren Zerstörung des Stators führen kann!
- Im Läufer der *LinMot® P* Linearmotoren befinden sich Neodym-Magnete, die bereits bei Annäherung magnetische Datenträger sowie empfindliche elektronische Geräte stören bzw. beschädigen können.
- Beim Manipulieren mit den Läufern ist ein “Anklatschen” an Eisenteile, Werkzeuge u.ä. unbedingt zu vermeiden, da dies zu Schäden am Läufer führt (Oberflächenbeschädigung, Verkrümmung). Das “Anklatschen” der Läufer an eisenhaltige Gegenstände stellt zudem eine Verletzungsgefahr dar (Quetschungen an Fingern etc.).
- Die Läufer der *LinMot® P* Motoren können Temperaturwerte erreichen, welche bei Berührung zu Brandverletzungen führen.
- Die Läufer der *LinMot® P* Motoren sind schnell bewegte Maschinenteile. Der Anwender hat konstruktiv alle notwendigen Massnahmen zum Schutz vor deren Berührung und der damit zusammenhängenden Gefährdung von Lebewesen zu treffen (Abdeckung, Berührungsschutz etc.).
- Eine Verschmutzung der Lagerstellen, insbesondere mit eisenhaltigen Spänen (Magnetanziehung!) oder ein Trockenlauf der Läufer, kann zu einer Beeinträchtigung der Lebensdauer führen.
- Die Läufer sind ausschliesslich in den dafür vorgesehenen Kunststoffbehältern (mit Kartoneinlage) oder bereits in *LinMot® P* Statoren eingeschoben und gesichert zu lagern und zu transportieren.
- Die Läufer müssen vor Schmutz (insbesondere Metallspäne etc.) geschützt werden.
- Maximale Lagertemperatur: 70 °C
- Sowohl Läufer als auch Stator sind bei Verschmutzung zu reinigen und anschliessend wieder einzufetten (siehe Kapitel Service).



G. Kontaktadressen

Schweiz

NTI AG
Haerdlistr. 15
CH-8957 Spreitenbach

Verkauf und Administration:	office@linmot.com
Tech. Support:	+41-(0)56-544 71 00
	support@linmot.com
Fax:	+41-(0)56-419 91 92
Web:	http://www.linmot.com

USA

LinMot, Inc.
5750 Townline Road
Elkhorn, WI53121

Sales and Aministration:	877-546-3270
	262-743-2555
Tech. Support:	877-804-0718
	262-743-1284
Fax:	800-463-8708
	262-723-6688
E-Mail:	us-sales@linmot.com
Web:	http://www.linmot-usa.com

Bitte besuchen Sie http://www.linmot.com/ um den Distributor in Ihrer Nähe zu finden.



Index

Ziffern

A

B

C

D









W

Z